Visual Anchors Are Strong Information Aggregators For Multimodal Large Language Model

Haogeng Liu[1,2], **Quanzeng You**[3], **Xiaotian Han**[3], **Yongfei Liu**[3],
Huaibo Huang[1,2]*, **Ran He**[1,2], **Hongxia Yang**

[1]MAIS & NLPR, Institute of Automation, Chinese Academy of Sciences
[2]School of Artifcial Intelligence, University of Chinese Academy of Sciences
[3]ByteDance, Inc
`liuhaogeng22@mails.ucas.ac.cn, huaibo.huang@cripac.ia.ac.cn`

Abstract

In the realm of Multimodal Large Language Models (MLLMs), vision-language connector plays a crucial role to link the pre-trained vision encoders with Large Language Models (LLMs). Despite its importance, the vision-language connector has been relatively less explored. In this study, we aim to propose a strong vision-language connector that enables MLLMs to achieve high accuracy while maintain low computation cost. We first reveal the existence of the visual anchors in Vision Transformer and propose a cost-effective search algorithm to extract them. Building on these findings, we introduce the Anchor Former (AcFormer), a novel vision-language connector designed to leverage the rich prior knowledge obtained from these visual anchors during pretraining, guiding the aggregation of information. Through extensive experimentation, we demonstrate that the proposed method significantly reduces computational costs by nearly two-thirds compared with baseline, while simultaneously outperforming baseline methods. This highlights the effectiveness and efficiency of AcFormer. Codes are available at https://github.com/liuhaogeng/Anchor-Former.

1 Introduction

Multimodal Large Language Models (MLLMs) have emerged as a focal point within contemporary research discourse[47]. Prominently showcased by seminal works such as LLaVA [33, 34], BLIP-2 [27], Qwen-VL [4], and Flamingo [1], these models exhibit exceptional efficacy across a broad spectrum of tasks, spanning from nuanced image description [29, 48] to complex visual reasoning. Their versatility transcends conventional boundaries, finding practical application in quotidian scenarios such as smartphone interface design [21] and consequential real-world decision-making processes [13]. This advancement is attributed to the availability of pre-trained Large Language Models (LLMs) [44] and vision encoders [41]. By utilizing these pre-trained components and introducing a connecting layer between them, it is possible to construct robust MLLMs with only training the lightweight vision-language connector. This enables the development of MLLMs with the capacity to process visual inputs while retaining the linguistic prowess characteristic of LLMs. These enhanced MLLMs exhibit proficiency across various tasks, including narrative generation, code composition, and addressing complex queries [45].

In the construction of above MLLMs, the vision-language connector plays a pivotal role. A fundamental approach, as demonstrated in LLaVA [34], employs a linear projection layer as the connector. In the enhanced version, LLaVA-1.5 [33], the linear projection layer is expanded into a multilayer

---

*Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).
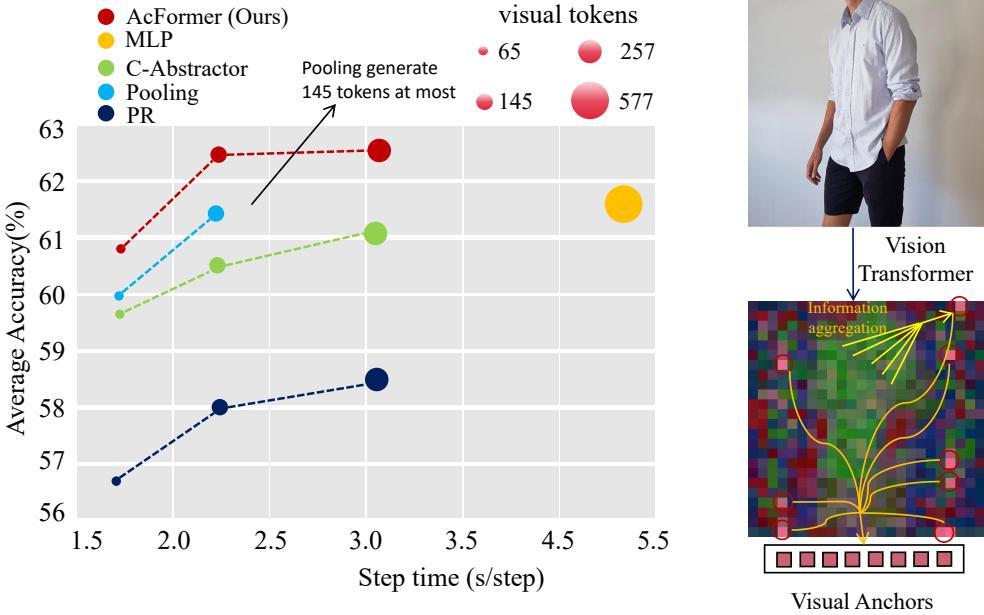
Figure 1: Comparison of the **average normalized accuracy** (MMB, TextVQA, GQA). PR means Perceiver Resampler, which utilize the learnable query as information aggregator. Our method achieves **highest accuracy** comparing with the others while maintaining **high training speed**.

perceptron, enhancing the model's efficacy. Despite achieving notable performance, the large number of visual tokens, which extends the time required for computing attention and other processes [8], indicating potential for optimization to decrease the computation cost. Prior efforts have sought to address this concern. For instance, BLIP-2 [27] introduces Q-Former and Flamingo [1] proposes Perceiver Resampler, both leveraging learnable queries as visual information aggregators. This mechanism utilizes the cross-attention between learnable queries and the outputs of visual encoders to effectively reduce the length of the visual sequence, thereby lowering the computation cost. Similarly, Qwen-VL [4] adopts a comparable structure but eliminates self-attention among the learnable queries. While these vision-language connectors substantially improve efficiency compared to naive linear projectors, they also exhibit a notable decrease in accuracy, as detailed in [5].

The primary computational bottleneck in MLLMs is the LLM component, where computational costs escalate significantly with the length of the input sequence, including both visual and textual tokens. To mitigate the computational cost, a straightforward approach is to reduce the number of input visual tokens with vision-language connector. A commonly adopted method is attention pooling, which offers greater flexibility than traditional pooling techniques. This method focuses on aggregating information within the visual tokens, necessitating aggregators with high information-gathering capabilities. Current attention pooling methods typically use randomly initialized learnable queries as information aggregators. We identify two main drawbacks with this approach: (1) The queries are randomly initialized without prior knowledge for aggregating visual information. They necessitates training on massive datasets (hundreds of millions) to be effective, as showed in Table 5 (2) The queries are fixed and invariant to different input images, potentially leading to significant information loss and low specificity for uncommon inputs, as illustrated in [22, 39].

To address the aforementioned issue, we propose Anchor Former (AcFormer), a novel vision-language connector that enhances both the accuracy and efficiency of MLLMs compared with baselines. In order to build AcFormer, we identify more effective information aggregators by analyzing the visual features obtained from a pre-trained vision encoder from two perspectives: the feature map and the attention map. Our analysis reveals the presence of "visual anchors" within the visual tokens. Fundamentally, the transformers in the neural network aggregate information related to these visual anchors, central to the transformation process. Moreover, the positions of these anchors vary across different images. Despite this variability, we can effectively identify them using the attention matrix to carry out the cost-effective progressively search algorithm. With these observations, we propose a

2

Anchor Selector, an important part of our AcFormer. The Anchor Selector utilize the progressive search algorithm with respect to the attention map. By this way, it effectively extracts visual anchors from the visual tokens generated by the Vision Transformer. And with the visual anchors, AcFormer utilize the naive cross attention to aggregate visual information for generating dense and complete visual representation.

In summary, our contributions can be summarized as follows:

- We reveal the existence of visual anchors within the visual tokens generated by pre-trained Vision Transformer, and subsequently propose cost-effective Anchor Selector to effectively extract these visual anchors.
- We propose the Anchor Former (AcFormer), a novel vision-language connector designed to improve the accuracy and efficiency of Multimodal Large Language Models (MLLMs) by leveraging the rich prior of information aggregation within visual anchors.
- We conduct comprehensive experiments across various vision-language tasks to empirically validate the efficacy of AcFormer.

2 Related Work

2.1 Multimodal Large Language Models

The development of MLLMs has become financially viable due to the utilization of pre-trained vision encoders [41, 38, 12] and Large Language Models (LLMs) [44, 43, 52]. Spearheaded by initiatives like Flamingo [1] and BLIP-2 [28], the field has witnessed significant advancements [34, 42, 53, 3, 51, 45, 21]. Studies like LLaVA and MiniGPT-4 have introduced methodologies such as visual instruction tuning, enabling robust MLLMs capable of understanding human instructions. Additionally, efforts such as Emu and LaVIT [24] have proposed unified frameworks for generation and comprehension, integrating visual decoders and consolidating the training loss of visual and textual inputs. The progression of MLLMs has been supported by the availability of extensive visual-language training datasets [7, 33]. Innovations like Sphinx [32] and Monkey [31] have facilitated high-resolution image processing through techniques like sub-image cropping, thus advancing open-source MLLMs. Moreover, MobileVLM [10] has introduced a compact vision language model suitable for deployment on mobile devices.

2.2 Vision-Language Connectors

Vision-language connectors typically employ either direct linear projection (LLaVA) or an information aggregation module, such as Flamingo, BLIP-2, and C-Abstractor, followed by linear projection. LLaVA, utilizing a simple multi-layer perceptron, effectively aligns visual features with the embedding space of Large Language Models (LLMs), but it suffers from high computational costs due to redundant input visual tokens, as highlighted by [8]. BLIP-2 uses the Q-Former to aggregate visual information and establish robust baselines, while Flamingo employs the Perceiver Resampler. Both architectures leverage the cross-attention mechanism to aggregate visual information into learnable queries. However, these approaches require extensive data for training and may have limitations in tasks requiring fine-grained visual perception due to the constrained nature of the learned query's ability to capture all visual patterns. In contrast, Honeybee [5] proposes the C-Abstractor and D-Abstractor to address these challenges by introducing spatial priors into the feature representation. Our proposed method, AcFormer, consists of three parts: Anchor Selector, Information Aggregation Module and Linear Projection. While Flamingo and BLIP-2 use learnable queries for information aggregation, C-Abstractor applies a convolution network directly. Our method utilizes visual anchors as information aggregators, generating dense and complete visual representations for input images.

3 Methods

3.1 Preliminaries

For MLLMs, their visual encoders are typically off-the-shelf pre-trained Vision Transformers (CLIP). Given input images $\mathbf{I} \in \mathbb{R}^{B \times C \times H \times W}$ and a Vision Transformer denoted as $F(\cdot)$, the vision feature

3
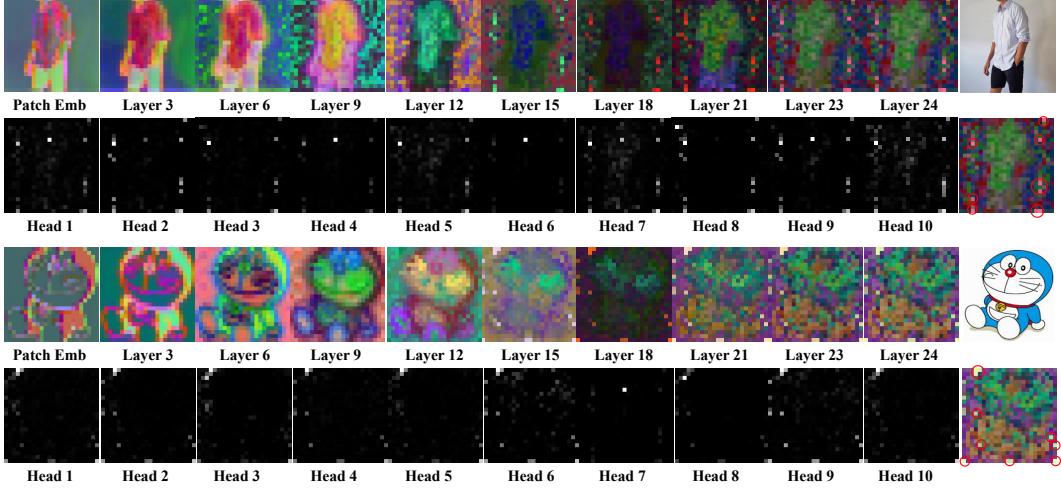
Figure 2: Visualizations of the visual feature map and attention map pertaining to the [CLS] token. Here we select 10 layers in Vision Transformer to show their output. We present the attention maps corresponding to the [CLS] token in the final layer. Notably, special tokens within both the feature map and attention map are identified using red circles. These marked points are referred to as "visual anchors". Details can be found in Section 3.2.

$\mathbf{R_v}$ is obtained as follows:

$$\mathbf{R}_v = F(\mathbf{I}) \in \mathbb{R}^{B \times N \times D}. \tag{1}$$

Here, C, H, and W represent the channel, height, and width of the input image, respectively, while N and D denote the number and dimension of the image tokens. There is one additional token, the [CLS] token added for the global representation of the image in contrastive learning. After obtaining the feature map, two common methods are used to combine it with LLM.

The first method utilizes gated cross-attention for modality fusion. Assuming the corresponding input instruction with the image is $\mathbf{T} \in \mathbb{R}^{B \times N_t \times D_t}$, the computation can be expressed as:

$$\mathbf{T}_h = G(\text{query} = \mathbf{T}, \text{key} = \mathbf{R}_v, \text{value} = \mathbf{R}_v), \tag{2}$$

where $\mathbf{T_h}$ represents the hidden states of the LLM and $G(\cdot)$ denotes the gated cross-attention layer.

The other approach involves converting the visual tokens into soft embeddings nd concatenating them with the text embeddings as the input of the LLM. Suppose the vision-language connector is V-L-Connector,

$$\mathbf{LM}_{in} = \text{Concat}(\text{Proj}(\text{V-L-Connector}(\mathbf{R}_v)), \mathbf{T}), \tag{3}$$

Where Proj means the Linear Projection and $\mathbf{T}$ means the text embedding. $\mathbf{LM}_{in}$ represents the input embeddings of the LLM.

3.2 Visual Anchors

Within the ViT, the input pixel-level features undergo a series of transformations. Understanding how the visual semantic is learned will bring us better insight of building the vision-language connector. To analyse in a more intuitive way, we visualize the feature map and attention map.

Given a set of Vision Transformer's feature maps, denoted as $\mathbf{V} \in \mathbb{R}^{N \times D}$, where N represents the number of tokens and D signifies the dimension of these tokens, we leverage dimension reduction for visual feature visualization. Initially, we extract all hidden states from the Vision Transformer. Subsequently, employing Principal Component Analysis (PCA) on each individual feature map, we derive low-dimensional features. We select the first three principal components to yield $\mathbf{V}' \in \mathbb{R}^{N \times 3}$, followed by normalizing the pixel values to the range of $[0, 255]$. To visually represent these features, we construct an image of the input size. Each patch within the image is then encoded using the obtained three-dimensional representation, effectively encapsulating the corresponding region's

4

value within the image. By this way, we obtain the visualization of the features, as depicted in Figure 2 (Row 1 and Row 3). Regarding the attention map, we derive it by extracting the attention weights associated with the last layer's [CLS] token. These attention scores reflect the significance of respective tokens. We exclude the attention directed towards the [CLS] token itself, yielding the attention map $\mathbf{A} \in \mathbb{R}^{H \times N}$, where H and N denote the number of attention heads and visual tokens. This process is depicted in Figure 2(Row 2 and Row 4).

Upon visual inspection, several noteworthy observations come to light. Specifically, the transformation of visual information exhibits a gradual obscuration of the feature map, accompanied by an increasing activation of specific tokens (depicted as pink and light yellow patches in the first and third lines, respectively). Initially, these activated tokens are just parts of the background, appearing indistinguishable from surrounding elements. However, over time, they progressively differentiate themselves from the background. Nevertheless, this evolutionary process lacks a discernible pattern. Examining the attention map, the [CLS] token, commonly used for aggregate global information of the input image, is anticipated to attend to the most salient regions. Paradoxically, the attention map of the [CLS] token predominantly focuses on a limited subset of tokens. We calculate the overlap among the activated tokens in feature map and the salient regions in attention map. The ratio reaches up to **69.38**% (500 images are sampled for calculation), conjecturing that this alignment is not coincidental. For further validation, we visualize the pre-trained MLLMs text generation attention matrix (text to visual tokens), result can be found in Figure 5.

Based on above observation, we name these tokens "visual anchors" and assume them serving as pivotal points for information aggregation during the transformation of visual features. While the [CLS] token indeed integrates visual information, its reliance alone proves insufficient, necessitating the involvement of other visual anchors in conveying information to the [CLS] token. Consequently, this process facilitates the extraction of meaningful representations from the image.
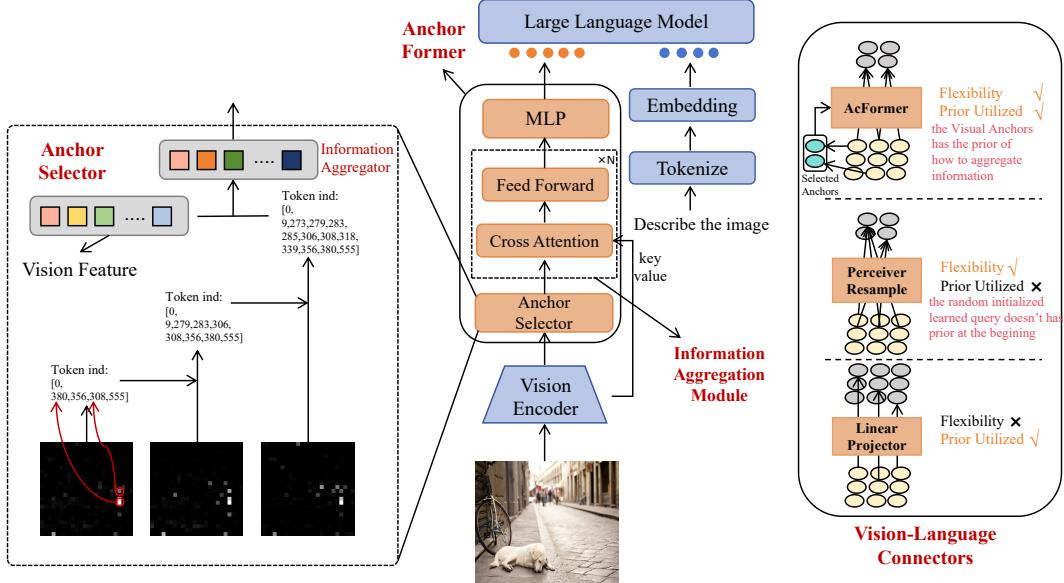
Figure 3: Visualization of Anchor Former (AcFormer). We propose our token selection algorithm code in detail at Section D.

3.3 Anchor Former

As illustrated above, the vision information is aggregated through visual anchors. Similarly, within Multimodal Large Language Models (MLLMs), structures such as Q-Former and Perceiver Resampler also leverage information aggregation modules. However, these models use learnable queries as Information Aggregator to extract information from the visual feature map produced by a pre-trained Vision Transformer. In this method, the same queries are used for all images, which can lead to two issues. First, their effectiveness requires extensive datasets for training as demonstrated in Table 5. Second, they may lead to significant information loss, as illustrated in [5, 39, 22].

5

To address the aforementioned issues, we propose Anchor Former (AcFormer), which consists of Anchor Selector, Information Aggregation Module and Linear Projector. We show a rough view of Anchor Former in Figure 3. Our approach integrates insights from visual anchors with the established framework of the Perceiver Resampler. A key innovation in our method is the use of visual anchors associated with each image as Information Aggregator for aggregating visual information. The effectiveness of our approach relies on the selection of these salient points. This visual anchors allows for more precise and specific information extraction, improving both the accuracy and efficiency of the model.

Anchor Selector. To avoid additional computation, we leverage the attention map of the [CLS] token for visual anchor selection. We introduce a progressive search algorithm to construct the Anchor Selector. Assuming the attention map is $\mathbf{A} \in \mathbb{R}^{H \times (N-1)}$, where H represents the number of attention heads and $N - 1$ denotes the number of visual tokens ([CLS] excluded). Let the token index list be TL, initially containing only the index 0, corresponding to the [CLS] token, an essential anchor. Suppose we still need T_N tokens in addition to the [CLS] token. We select these tokens head by head, assuming each head will provide $\frac{T_N}{H}$ tokens. For each head, , we first sort the indices of the visual tokens based on their attention scores. Then, we select the top $\frac{T_N}{H}$ tokens from this sorted list. If the chosen token is already in TL, we choose the next token in the sorted order until we have the required number of unique tokens. We provide detailed algorithm in Figure 7.

Information Aggregation Module. In our approach, we employ selected visual anchors as Information Aggregator, combined with a cross-attention module to aggregate information. Let the Information Aggregator be denoted as $\mathbf{IA} \in \mathbb{R}^{(T_N+1) \times D}$ and the origin visual tokens as $\mathbf{R}_v \in \mathbb{R}^{N \times D}$. Our proposed model, Information Aggregation Module, is a bidirectional transformer encoder. We denote the cross-attention module as Attn and the feedforward module as FF. Let LN represent layer normalization. For a single layer in the model, the operations are as follows:

$$\mathbf{A}_{out} = \mathbf{IA} + \text{Attn}(\text{query=LN}(\mathbf{IA}), \text{key=LN}(\mathbf{R}_v), \text{value=LN}(\mathbf{R}_v)), \tag{4}$$

$$\mathbf{H} = \mathbf{A}_{out} + \text{FF}(\text{LN}(\mathbf{A}_{out})), \tag{5}$$

Where $\mathbf{H}$ means the hidden states. We use $\mathbf{H}_v$ to represent the last hidden states. Let $\mathbf{T}$ represents text embedding. We use the Proj to represent the Multilayer Perception. We obtain the final multimodal input embeddings for LLM as bellow,

$$\mathbf{LM}_{in} = \text{Concat}(\text{Proj}(\mathbf{H}_v), \mathbf{T}). \tag{6}$$

4 Experiments

4.1 Settings

Benchmarks. We employ nine distinct benchmarks to comprehensively assess the overall efficacy of our proposed method. The specifics of these benchmarks are delineated in Table 7. Notably, in our experimental setup, as we reduce the number of image tokens, our focus is primarily directed towards enhancing visual perception capabilities. As a result, we pay particular attention to benchmarks such as TextVQA, which challenged the model's fine-grained visual perception ability [21, 31].

Implementation details. In our experimental setup, we utilize 7B and 13B Vicuna-v1.5 as Large Language Models (LLMs) [9]. The CLIP ViT-L/14 model, pre-trained with a resolution of 336, serves as our vision encoder. We select the last but one layer's output from the vision encoder as our vision feature. For Anchor Former, we configure it with 6 layers and a hidden dimension of 512, employing 8 attention heads, each with a dimension of 64. The feedforward module utilize 2048 as the hidden dimension. Regarding the training dataset, we leverage the dataset utilized in LLaVA-1.5 [33], with 558k samples for pre-training and 665k samples for instruction tuning. Our experimentation encompasses the evaluation of various vision-language connectors, including the Perceiver Resampler, Anchor Former, pooling, C-Abstractor and utilizing pooled tokens as queries for the Perceiver Resampler. To maintain consistency, we construct our model using the official code provided by LLaVA. Specifically, we adjust the pre-training initial learning rate from $1e^{-3}$ to $5e^{-4}$. Other configuration is the same with origin LLaVA-1.5.

6

Table 1: Results on benchmark designed for MLLMs. V-T Num means the visual tokens number. V-T Num influences the computation cost that the bigger the V-T Num the heavier the computation cost is. Speed here means the relative pre-training speed with respect to LLaVA-1.5.

Model	LLM	Connector	V-T Num	Res	POPE	MME	MMB	MM-Vet	Speed (↑)
Approaches using 7B Large Language Models									
MiniGPT-4 [53]	Vicuna-7B	Resampler	32	224	72.2	726.0	24.3	22.1	-
mPLUG-Owl2[46]	LLaMA2-7B	Resampler	32	224	-	1243.4	49.4	-	-
InstructBLIP[11]	LLaMA2-7B	Q-Former	32	224	78.9	-	36.0	26.2	-
LLaVA (v1) [34]	LLaMA-7B	Linear	257	224	67.7	717.5	38.7	-	-
LLaMA-AdapterV2 [16]	LLaMA2-7B	LLaMA-Adapter	257	224	-	1221.6	41.0	31.4	-
Shikra [6]	Vicuna-7B	Linear	257	224	-	-	58.8	-	-
Qwen-VL[4]	Qwen-7B	Resampler	256	448	-	-	38.2	-	-
Qwen-VL-Chat[4]	Qwen-7B	Resampler	256	448	-	1845.3	60.6	-	-
LLaVA-1.5 [33]	Vicuna-7B	Linear	577	336	85.9	1794.6	64.3	**30.5**	1.00×
Ours	Vicuna-7B	AcFormer	145	336	**86.4**	**1846.1**	**68.4**	30.3	**2.23×**
Approaches using 13B Large Language Models									
MiniGPT-4 [53]	Vicuna-13B	Resampler	32	224	-	1158.7	-	24.4	-
InstructBLIP[11]	Vicuna-13B	Q-Former	32	224	78.9	1504.6	-	25.6	-
BLIP-2[28]	Vicuna-13B	Q-Former	32	224	85.3	-	-	22.4	-
LLaVA-1.5 [33]	Vicuna-13B	Linear	577	336	85.9	1826.7	67.7	**35.4**	1.00×
Ours	Vicuna-13B	AcFormer	145	336	**86.1**	**1870.0**	**69.2**	34.1	**2.30×**

4.2 Main Results

We present the main results in Tables 1 and 2, organized by benchmark type. Upon observation of these tables, it becomes apparent that our model achieves robust performance despite being trained with limited data and visual tokens. Notably, even with only 145 or 257 tokens, our model achieves performance comparable to that of the original LLaVA-1.5 model, which utilizes 577 visual tokens as input. This performance holds across various benchmarks, including those that require high-level visual perception (e.g., VisWiz, TextVQA) and those that assess overall capability (e.g., MME, GQA).

However, it should be noted that although our model is overall effective, it performs slightly worse than LLaVA-1.5 on certain benchmarks such as GQA and VQAv2. Given that our method only applies significantly fewer visual tokens (less than half), the slightly performance drop meets expectations. Nevertheless, the performance gap is relatively small and can be considered marginal in light of the substantial increase in speed.

Table 2: Results on General VQA tasks. V-T Num means the visual tokens number. V-T Num influences the computation cost that the bigger the V-T Num the heavier the computation cost is. Speed here means the relative pre-training speed with respect to LLaVA-1.5.

Model	LLM	Connector	V-T Num	Res	TextVQA	GQA	VQAv2	VisWiz	SQA$_{img}$	Speed (↑)
Approaches using 7B Large Language Models										
InstructBLIP[11]	LLaMA2-7B	Q-Former	32	224	-	49.2	-	34.5	60.5	-
Shikra [6]	Vicuna-7B	Linear	257	224	-	-	77.4	-	-	-
IDEFICS-9B [25]	LLaMA-7B	Cross Attn	257	224	-	38.4	50.9	35.5	-	-
Qwen-VL[4]	Qwen-7B	Resampler	256	448	-	59.3	**78.8**	35.2	67.1	-
Qwen-VL-Chat[4]	Qwen-7B	Resampler	256	448	-	57.5	78.2	38.9	68.2	-
LLaVA-1.5 [33]	Vicuna-7B	Linear	577	336	58.2	**62.0**	78.5	50.0	66.8	1.00×
Ours	Vicuna-7B	AcFormer	257	336	**58.2**	61.2	78.4	**52.8**	**69.4**	**1.65×**
Approaches using 13B Large Language Models										
InstructBLIP[11]	Vicuna-13B	Q-Former	32	224	-	49.5	-	33.4	63.1	-
BLIP-2[28]	Vicuna-13B	Q-Former	32	224	-	41.0	41.0	19.5	61.0	-
LLaVA-1.5 [33]	Vicuna-13B	Linear	577	336	61.2	**63.3**	**80.0**	53.6	71.6	1.00×
Ours	Vicuna-13B	AcFormer	257	336	**61.3**	63.0	79.8	**53.7**	**71.8**	**1.69×**

4.3 Ablation Results

We mainly compare different visual Connectors. To facilitate understanding, we provide definitions for some terms. Pooling denotes direct pooling of visual token. Pooling-PR employs the pooled

7

Table 3: Ablation studies. "Pooling" denotes direct pooling of visual token. "Pooling-PR" employs the pooled tokens as queries for the Perceiver Resampler. "Random-PR" means the Perceiver Resampler using randomly selected tokens from the vision feature map as query. "PR" refers to the Perceiver Resampler using learnable queries. "AcFormer" represents our proposed Anchor Former. The configuration of the C-Abstractor follows Honeybee [5]. V-T Num means the visual tokens number.

Model	LLM	Connector	V-T Num.	TextVQA	GQA	MMB	MME
LLaVA-1.5	Vicuna-7B	Pooling	65	53.4	59.8	66.8	1734.0
	Vicuna-7B	Pooling-PR	65	53.9	**60.0**	66.8	1728.9
	Vicuna-7B	Random-PR	65	53.9	59.1	66.9	1728.7
	Vicuna-7B	PR	65	51.0	56.1	63.2	1702.8
	Vicuna-7B	C-Abstractor	65	52.8	59.0	67.0	1743.3
	Vicuna-7B	AcFormer	65	**56.1**	59.2	**67.3**	**1744.2**
LLaVA-1.5	Vicuna-7B	Pooling	145	55.1	60.9	68.0	1791.4
	Vicuna-7B	Pooling-PR	145	54.7	60.9	68.0	1759.1
	Vicuna-7B	Random-PR	145	54.6	59.7	67.0	1772.7
	Vicuna-7B	PR	145	52.1	56.4	65.4	1720.8
	Vicuna-7B	C-Abstractor	145	53.4	60.2	67.8	1775.4
	Vicuna-7B	AcFormer	145	**58.0**	**61.3**	**68.4**	**1846.1**
LLaVA-1.5	Vicuna-7B	PR	257	52.3	56.8	65.7	1735.9
	Vicuna-7B	C-Abstractor	257	53.7	60.8	68.3	1790.0
	Vicuna-7B	AcFormer	257	**58.2**	**61.2**	**68.3**	**1848.8**
LLaVA-1.5	Vicuna-13B	PR	145	53.4	56.9	64.7	1749.3
	Vicuna-13B	C-Abstractor	145	58.5	62.1	68.8	1823.6
	Vicuna-13B	AcFormer	145	**60.7**	**62.8**	**69.2**	**1869.3**

Table 4: Ablation studies on whether to directly use the selected tokens as input.

Model	LLM	Connector	V-T Num.	TextVQA	GQA	MMB	MME
LLaVA-1.5	Vicuna-7B	Top-P	145	56.3	60.8	68.2	1798.8
	Vicuna-7B	E-ViT	146	57.1	61.0	68.3	1808.4
	Vicuna-7B	AcFormer	145	**58.0**	**61.3**	**68.4**	**1846.1**

tokens as queries for the Perceiver Resampler. Random-PR means the Perceiver Resampler using randomly selected tokens from the vision feature map as query. PR refers to the commonly used Perceiver Resampler. Our experimental findings validate the efficacy of our model from multiple perspectives. A comparison between PR and AcFormer indicates that the observed enhancement does not stem solely from an increase in **trainable parameters**. Additionally, comparison between AcFormer and Random-PR underscores the critical role of **Anchor Selector** in model performance. Furthermore, our evaluation of AcFormer against C-Abstractor reveals that the significance of inductive bias diminishes, as spatial coherence can be effectively maintained through cross-attention mechanisms within the Perceiver Resampler.

C-Abstractor. We trained the model with training data from LLaVA-1.5. Our findings, as presented in Table 3, indicate that the C-Abstractor method achieves comparable performance to AcFormer across most benchmarks. However, in tasks such as TextVQA, which necessitates high-level visual perception (often demanding fine-grained visual analysis), C-Abstractor exhibits inferior performance. This empirical evidence underscores the efficacy of our proposed AcFormer.

Pooling. One immediate consideration is to aggregate visual tokens based on their spatial positions by pooling and combine them with the [CLS] token to form the input visual features. Our empirical investigation reveals that direct pooling emerges as a potent technique for compressing visual information. However, it fails in TextVQA, leading to around 3 points drop.

Pooling-PR. Given our direct utilization of pooled tokens within Large Language Models, a concern arises regarding whether the observed degradation stems from the reduction in trainable parameters within the Anchor Former. To address this concern, we conducted additional experiments wherein the pooled tokens served as queries for the Perceiver Resampler, termed as Pooling-PR. Examination

8

of the results in the table reveals that this approach yielded even poorer performance compared to directly inputting the pooled tokens.

Random-PR. While our method demonstrates superior performance compared with other approaches, a lingering question pertains to whether this improvement indeed stems from the Anchor Former. To address the concern, we randomly select tokens from the vision feature map as the Information Aggregator. The empirical results presented in Table 3 corroborate the efficacy of our proposed anchor selection method, substantiating its discernible impact on model performance.

PR. While substantiated the effectiveness of AcFormer, an unresolved question pertains to its superiority over the commonly used Perceiver Resampler, which employs randomly initiallized learnable queries for visual information aggregation. Examination of the table reveals that the Perceiver Resampler exhibits the poorest performance among the aforementioned methods. This observation underscores the diminished performance of the structure under conditions of limited training data.

Anchor Direct-in. A related study [20] employ Top-K selection methodology for token compression. This approach shares similar token selection method with ours and involves the direct utilization of selected tokens as representations (without further cross attention) for visual information. However, it primarily caters to classification tasks, which prioritize global understanding, potentially rendering it less suitable for tasks requiring nuanced comprehension. Noteworthy is the proposition by the authors of Haurum [20] regarding EViT. They employ the selected tokens and incorporate pooling mechanisms for the remaining tokens as input. The findings, as detailed in Table 4, empirically underscore the significance of retaining unselected tokens, as they still encapsulate valuable information.

Table 5: Ablation studies on the visual connector when scaling up the training data.

Model	LLM	Connector	V-T Num.	TextVQA	GQA	OKVQA	VQAv$_2$	VizWiz	MME
Pretrain Dataset: 60M image-text pairs from LAION-115M, COYO, LAION COCO **Instruction Finetuning Dataset**:LLaVA-665k									
LLaVA-1.5	OpenLLaMA-3B	PR	145	35.03	54.35	54.14	70.04	31.16	1592.7
	OpenLLaMA 3B	AcFormer	145	**35.89**	**55.45**	**55.15**	**72.76**	**33.88**	**1622.3**
Pretrain Dataset: 60M image-text pairs from LAION-115M, COYO, LAION COCO **Instruction Finetuning Dataset**:Cauldron (roughly 1.8M)									
LLaVA-1.5	OpenLLaMA-3B	PR	145	38.76	44.98	49.66	70.83	32.87	1502.5
	OpenLLaMA 3B	AcFormer	145	**40.49**	**45.67**	**50.57**	**73.01**	**32.97**	**1523.1**

4.4 Scaling Up the Training Dataset

We conducted a data scaling experiment to further evaluating our proposed method's performance on large-scale data scene. Considering the expensive training cost, we replaced the Large Language Model with OpenLLaMA-3B [17]. Approximately 60 million image-text pairs were employed for pre-training, sampled from Laion115M, coyo-238M, and laion-coco100M datasets. For instruction fine-tuning, we utilized two datasets, covering LLaVA-665k and Cauldron [26]. Notably, Cauldron (1.8M) is a larger instruction finetuning dataset than LLaVA-665k (0.66M) for MLLMs.

Our focus was on comparing Perceiver Resampler with our proposed AcFormer. The results, summarized in Table 5, consistently demonstrate the superiority of AcFormer over the Perceiver Resampler. This illustrate that our method not only works well for the scene of limited data but also for larger data (both pre-train dataset and instruction tuning dataset).

5 Conclusions

In this study, we reveal the existence of the visual anchors and present the hypothesis that visual anchors are strong visual information aggregators. With the observation, we propose the Anchor Former, an effective vision-language connector. Notably, Anchor Former distinguishes itself from the conventional information aggregation methods (e.g., Q-Former and Perceiver Resampler) by adopting

9

visual anchors as Information Aggregator. To build Anchor Former, we present the progressive search algorithm to effectively extract the visual anchors. Extensive experiments on different benchmarks consistently underscores the efficacy of Anchor Former in improving the models' accuracy while simultaneously removing the redundant visual tokens in MLLMs.

Acknowledgements

This research is partially funded by National Natural Science Foundation of China (Grant No U21B2045, U20A20223), Beijing Nova Program (20230484276), and Youth Innovation Promotion Association CAS (Grant No. 2022132).

10

References

[1] 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, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc., 2022.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[3] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

[5] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. *arXiv preprint arXiv:2312.06742*, 2023.

[6] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic, 2023.

[7] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions, 2023.

[8] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models, 2024.

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[10] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023.

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.

[12] 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.

[13] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[14] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2023.

[15] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models, 2024.

[16] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. Llama-adapter v2: Parameter-efficient visual instruction model, 2023.

[17] Xinyang Geng and Hao Liu. Openllama: An open reproduction of llama, May 2023.

[18] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.

[19] Xiaotian Han, Quanzeng You, Yongfei Liu, Wentao Chen, Huangjie Zheng, Khalil Mrini, Xudong Lin, Yiqi Wang, Bohan Zhai, Jianbo Yuan, et al. Infimm-eval: Complex open-ended reasoning evaluation for multi-modal large language models. *arXiv e-prints*, pages arXiv–2311, 2023.

[20] Joakim Bruslund Haurum, Sergio Escalera, Graham W Taylor, and Thomas B Moeslund. Which tokens to use? investigating token reduction in vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 773–783, 2023.

[21] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

11

[22] Yi-Xin Huang, Hou-I Liu, Hong-Han Shuai, and Wen-Huang Cheng. Dq-detr: Detr with dynamic query for tiny object detection. *arXiv preprint arXiv:2404.03507*, 2024.

[23] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[24] Yang Jin, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, et al. Unified language-vision pretraining with dynamic discrete visual tokenization. *arXiv preprint arXiv:2309.04669*, 2023.

[25] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models?, 2024.

[27] 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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 23–29 Jul 2023.

[28] 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.

[29] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[30] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

[31] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*, 2023.

[32] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

[33] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[35] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.

[36] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[37] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[39] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, Yuda Xiong, Hao Zhang, Feng Li, Peijun Tang, Kent Yu, and Lei Zhang. Grounding dino 1.5: Advance the "edge" of open-set object detection, 2024.

[40] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

[41] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[42] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.

[43] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

12

[44] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[45] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

[46] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl: Modularization empowers large language models with multimodality, 2024.

[47] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models, 2024.

[48] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[49] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities, 2023.

[50] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

[51] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023.

[52] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[53] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023.

13

A   Visualization of The Feature From OpenAI CLIP

We have shown the visualization of the feature map and attention map of EVA-CLIP-L in the main text. To validate whether other models (OpenAI-CLIP) also has the same characteristic, we offer analogous visualizations in Figure 4, revealing comparable phenomena. While diverging from EVA-CLIP-L in certain aspects, OpenAI-CLIP-L nevertheless manifests certain anchor points. These shows that our observation is not an isolated case.
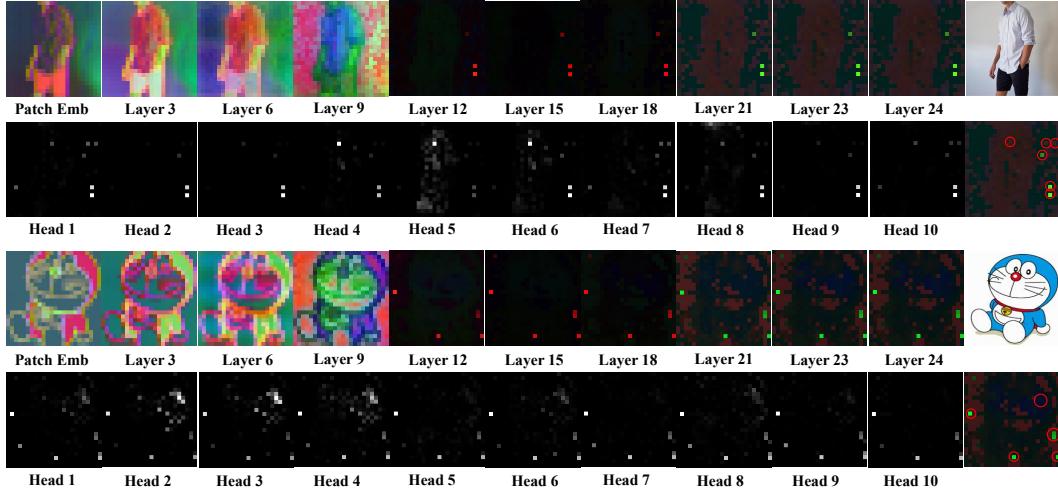
Figure 4: Visualization of the Openai CLIP. Figure 2 shows the visualization of EVA CLIP. We show that the phenomenon happens in different CLIP.

B   Visualization of The Attention Map From The Pre-trained MLLMs

To further substantiate our hypothesis, we conducted visualizations of the attention matrix of the pre-trained Multimodal Language Models (MLLMs). We believe directly obtaining the generated text's attention will give us better understanding of the visual anchors. Recognizing the potential generation of multiple tokens simultaneously, we adopted a methodology akin to LLaVA's approach, constraining the model to provide single-word or phrase responses for visualization of the attention mechanism on the answer (key word). Given the multilayered architecture of MLLMs, we specifically chose middle layers for visualization, as prior work [8] has indicated their pivotal role in comprehending visual signals.

Two types of MLLMs are considered, exemplified by Flamingo and LLaVA. As Flamingo segregates attention between text-only and image-text modalities, we opted for this model for visualization purposes. It is noteworthy that we excluded the Perceiver Resampler and directly leveraged all visual tokens to discern their relative importance. The visualization is accessible in Figure 5.

C   More Samples For Visual Anchor Visualization

We provide more visualization samples in Figure 6 to demonstrate that the phenomenon of vision anchors exists in both complex real-world scenes and animated scenes.

D   Detailed Algorithm

We have provided a rough description in Figure 3. Here, we present the detailed Python code for token selection in Figure 7. The complete code is available in the supplementary material. We use a top-k search to select visual tokens. Specifically, assume the attention matrix is $\mathbf{A} \in \mathbb{R}^{H \times 1 \times N}$, where H is the number of heads and N is the number of image patches. We traverse each head of the attention matrix, and for each head, we select $\frac{\mathbf{T}-1}{\mathbf{H}}$ tokens based on the sorted indices and already selected sequence to avoid duplication. Here, $\mathbf{T}$ represents the total number of desired tokens.

14
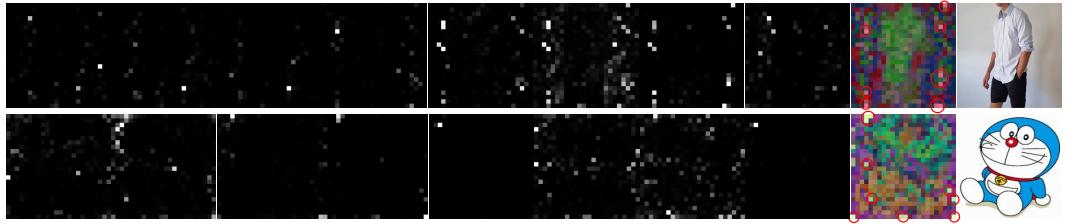
Figure 5: Visualization of the attention map from the pre-trained Flamingo model (Removing the Perceiver Resampler). The attention map is the generated text's attending to the corresponding image patches.
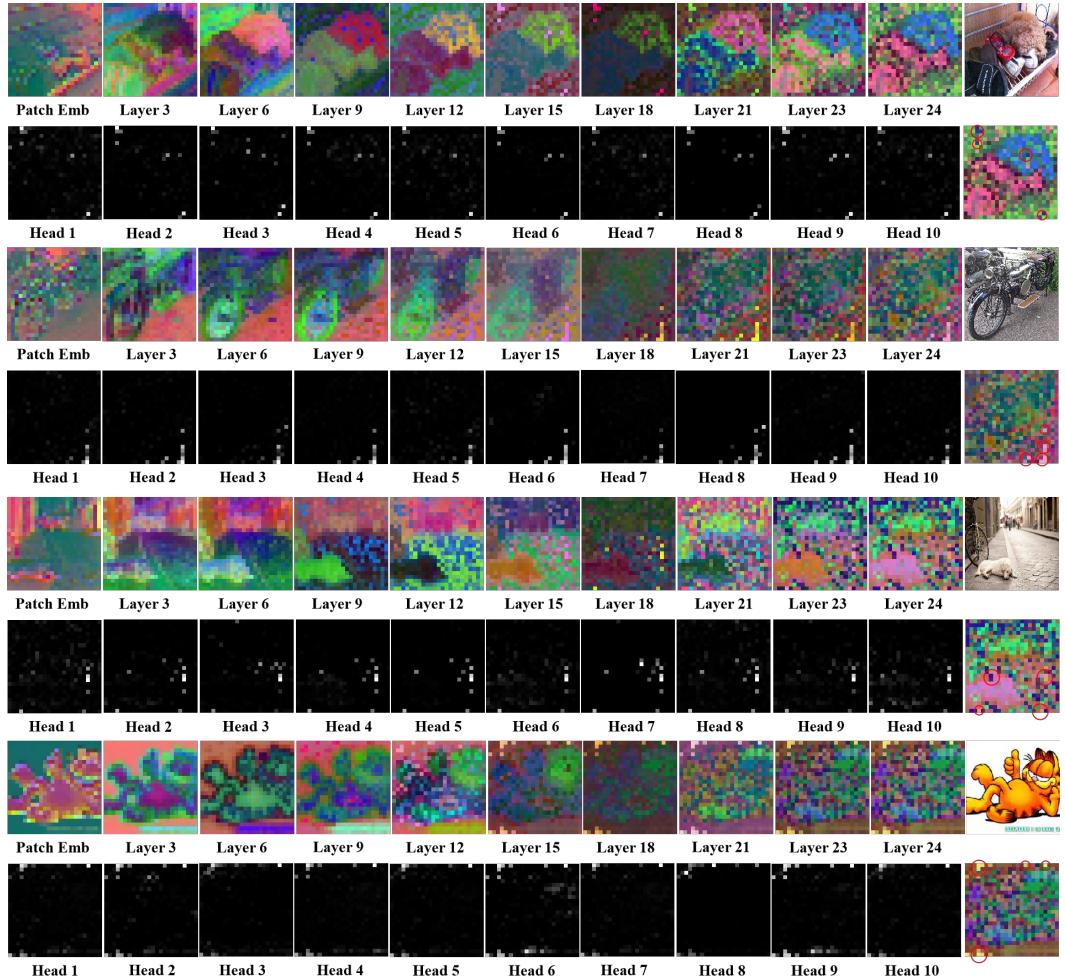
Figure 6: More samples for visualization of the visual anchors.

15

```
Algorithm: Anchor Selection
Input:
Visual Feature Map V (B, N, D)
Visual Attention Map A (B, H, N, N)
Token Number T
Caculation:
selected_anchor = None
Per_head_num = int((T-1) / H) #[CLS] is choosen by default
for i in range(B):
  max_indice = [0]
  for j in range(H):
    tmp_attn = A[i, j, 0, 1:]
    ind_sorted = Argsort(tmp_attn) + 1
    tmp_res = set(ind_sorted[-Per_head_num:] + max_indice)
    count = 1
    while len(tmp_res) < ((j+1) × per_head_num + 1) :
      tmp_res = set(ind_sorted[-Per_head_num-count:] + max_indice)
      count = count + 1
    max_indices = sorted(list(tmp_res))
    if selected_anchor is not None:
      selected_anchor = torch.cat((selected_anchor, V[[i], max_indices, :]), dim=0)
    else:
      selected_anchor = V[[i], max_indices, :]
Return:
selected_anchor
```

Figure 7: Detailed code for the anchor selector within Anchor Former (AcFormer).

Table 6: Details on the training time. Pt Bz means the pre-train batch size. And IFT Bz means the instruction finetune batch size.

Methods	LLM	Training resource	Visual Token Num	Pt Bz	IFT Bz	Pt time	IFT time
Pooling	Vicuna-7B	8 A100(80G)	65	256	128	39m	9h25m
Pooling-PR	Vicuna-7B	8 A100(80G)	65	256	128	42m	9h51m
Random-PR	Vicuna-7B	8 A100(80G)	65	256	128	42m	9h52m
Origin-PR	Vicuna-7B	8 A100(80G)	65	256	128	42m	9h52m
C-Abstractor	Vicuna-7B	8 A100(80G)	65	256	128	41m	9h51m
AcFormer	Vicuna-7B	8 A100(80G)	65	256	128	42m	9h52m
Pooling	Vicuna-7B	8 A100(80G)	145	256	128	1h03m	10h13m
Pooling-PR	Vicuna-7B	8 A100(80G)	145	256	128	1h20m	10h51m
Random-PR	Vicuna-7B	8 A100(80G)	145	256	128	1h19m	10h50m
Origin-PR	Vicuna-7B	8 A100(80G)	145	256	128	1h20m	10h52m
C-Abstractor	Vicuna-7B	8 A100(80G)	145	256	128	1h18m	10h50m
AcFormer	Vicuna-7B	8 A100(80G)	145	256	128	1h20m	10h12m
Origin-PR	Vicuna-7B	8 A100(80G)	257	256	128	1h50m	10h29m
C-Abstractor	Vicuna-7B	8 A100(80G)	257	256	128	1h49m	10h25m
AcFormer	Vicuna-7B	8 A100(80G)	257	256	128	1h50m	10h30m
Origin-PR	Vicuna-13B	8 A100(80G)	145	256	128	2h04m	17h31m
C-Abstractor	Vicuna-13B	8 A100(80G)	145	256	128	2h02m	17h29m
AcFormer	Vicuna-13B	8 A100(80G)	145	256	128	2h04m	17h32m

E   Training Resources

We provide details for our training resource for main experiment in Table 6. Notably, for the data scaling experiment, we utilize 16 Nvidia A100 (80G). It takes roughly 28 hours for pre-train and 5 hours for IFT.

F   Detailed Description of The Benchmarks

There exist numerous benchmarks for assessing the proficiency of Multi-Modal Large Language Models (MLLMs), each imbued with its own inherent biases. For instance, benchmarks such as OK-VQA [37] primarily concentrate on appraising the model's pre-existing knowledge base. Conversely,

16

Table 7: Details on the chosen benchmark.

Benchmark	Description of the task	Metric
TextVQA [40]	QAs about text in image (Visual Perception)	VQA score (↑)
VizWiz VQA [18]	QAs about image from blinds (Visual Perception)	VQA score (↑)
GQA [23]	QAs of real world comprehension and complex reasoning	EM (↑)
VQAv2 [2]	QAs require vision, language and prior world knowledge	VQA score (↑)
POPE [30]	QAs for Object Hallucination evaluation	F1 Score (↑)
Sci-QA(Img) [36]	QAs about Science	Accuracy (↑)
MME [14]	Comprehensive Evaluation Benchmark for MLLMs	Accuracy (↑)
MMbench [35]	Comprehensive Evaluation Benchmark for MLLMs	Accuracy (↑)
MM-Vet [49]	Integrated Capabilities Benchmark for MLLMs	GPT-4 score(↑)

benchmarks like TextVQA [40] and VizWiz-VQA [18] scrutinize the models' prowess in visual perception. Notably, a plethora of newly introduced benchmarks tailored specifically for MLLMs have surfaced. MME [15] and MMBench [35] stand out as comprehensive benchmarks aimed at gauging the overall performance of MLLMs. However, they mandate MLLMs to furnish responses in binary or multiple-choice formats. POPE [30] has been devised to assess MLLMs' propensity for hallucination. MM-VET [50] and InfiMM [19] scrutinize the open-ended question-answering capability of MLLMs, aided by the assistance of GPT-4.

G Limitations

Although we have conducted extensive experiments, there are still aspects requiring further investigation. For instance, the utilization of larger training datasets with corresponding larger models remains unexplored due to resource constraints in our experiments. Additionally, more theoretical analysis are needed for better elucidating the underlying reasons for the emergence of these visual anchors.

H Broader Impacts

Our model, despite its capabilities, may encounter certain risks. As it is built upon LLaMA, Vicuna, and CLIP, it inherits some issues associated with large language models (LLMs) and vision encoders. One significant risk is **hallucination**, where the model might generate content that contradicts the facts. This poses a concern, especially when applied in critical fields such as medicine. Additionally, **biases** present in the LLM and vision encoder (CLIP) could be transferred to our model, potentially resulting in biased outputs. Lastly, while **energy consumption** is not a primary concern due to the smaller pretraining dataset used, it may become an issue when scaling up the pretraining dataset or increasing the model size.

17

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We give main claims accurately in abstract and instruction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations at section G.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [No]

18

Justification: This is an **observational and experimentally validated study**, with no theoretical argumentation involved.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This work can be fully reproduced with the information from the paper. We provide the main code in the appendix 7 in detail.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

19

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our model is built based on the open-sourced LLaVA, so our provided code in Figure 7 is enough for reproduction of the method.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the implementation details in paragraph 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We experiment on many different benchmarks to eliminate statistical error.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

20

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

    Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

    Answer: [Yes]

    Justification: We report the training resources in section Table 6.

    Guidelines:

    - The answer NA means that the paper does not include experiments.
    - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
    - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
    - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

    Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

    Answer: [Yes]

    Justification: There is no bias or other special about the research.

    Guidelines:

    - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
    - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
    - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We provide Broader Impacts in section H.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

21

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: Our proposed method is easy for reproduct that we will not release the model.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We follow the license of the data and code used in our experiment.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

22

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: There are not any asset in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

23