ALIGNVLM: Bridging Vision and Language Latent Spaces for Multimodal Document Understanding

Ahmed Masry[1,2], **Juan A. Rodriguez**[1,3,4], **Tianyu Zhang**[1,3,5], **Suyuchen Wang**[1,3,5],
Chao Wang[1], **Aarash Feizi**[1,3,6], **Akshay Kalkunte Suresh**[1], **Abhay Puri**[1],
Xiangru Jian[1,7], **Pierre-André Noël**[1], **Sathwik Tejaswi Madhusudhan**[1],
Marco Pedersoli[1,4], **Bang Liu**[1,5,8], **Nicolas Chapados**[1], **Yoshua Bengio**[3,5,8],
Enamul Hoque[2], **Christopher Pal**[1,3,8,9], **Issam H. Laradji**[1,10], **David Vazquez**[1],
Perouz Taslakian[1], **Spandana Gella**[1], **Sai Rajeswar**[1,3,5]

[1]ServiceNow [2]York University [3]Mila – Quebec AI Institute
[4]École de Technologie Supérieure [5]Université de Montréal
[6]McGill University [7]University of Waterloo [8]CIFAR AI Chair
[9]Polytechnique Montréal [10]University of British Columbia

Abstract

Aligning visual features with language embeddings is a key challenge in vision-language models (VLMs). The performance of such models hinges on having a good connector that maps visual features generated by a vision encoder to a shared embedding space with the LLM while preserving semantic similarity. Existing connectors, such as multilayer perceptrons (MLPs), lack inductive bias to constrain visual features within the linguistic structure of the LLM's embedding space, making them data-hungry and prone to cross-modal misalignment. In this work, we propose a novel vision-text alignment method, ALIGNVLM, that maps visual features to a weighted average of LLM text embeddings. Our approach leverages the linguistic priors encoded by the LLM to ensure that visual features are mapped to regions of the space that the LLM can effectively interpret. ALIGNVLM is particularly effective for document understanding tasks, where visual and textual modalities are highly correlated. Our extensive experiments show that ALIGNVLM achieves state-of-the-art performance compared to prior alignment methods, with larger gains on document understanding tasks and under low-resource setups. We provide further analysis demonstrating its efficiency and robustness to noise.

1 Introduction

Vision-Language Models (VLMs) have gained significant traction in recent years as a powerful framework for multimodal document understanding tasks that involve interpreting both the visual and textual contents of scanned documents [Kim et al., 2022, Lee et al., 2023, Liu et al., 2023a, 2024, Hu et al., 2024, Wang et al., 2023a, Rodriguez et al., 2024b]. Such tasks are common in real-world commercial applications, including invoice parsing [Park et al., 2019], form reading [Jaume et al., 2019], and document question answering [Mathew et al., 2021b]. VLM architectures typically consist of three components: *(i)* a vision encoder to process raw images, *(ii)* a Large Language Model (LLM) pre-trained on text, and *(iii)* a connector module that maps the visual features from the vision encoder into the LLM's semantic space.

A central challenge in this pipeline is to effectively map the continuous feature embeddings of the vision encoder into the latent space of the LLM while preserving the semantic properties of visual concepts. Existing approaches can be broadly categorized into *deep fusion* and *shallow fusion* methods. *Deep fusion* methods, such as NVLM [Dai et al., 2024], Flamingo [Alayrac et al., 2022],

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

CogVLM [Wang et al., 2023b], and LLama 3.2-Vision [Grattafiori et al., 2024], integrate visual and textual features by introducing additional cross-attention and feed-forward layers at each layer of the LLM. While effective at enhancing cross-modal interaction, these methods substantially increase the parameter count of the VLM compared to the base LLM, resulting in high computational overhead and reduced efficiency.

In contrast, *shallow fusion* methods project visual features from the vision encoder into the LLM input embedding space using either multilayer perceptrons (MLPs) [Liu et al., 2023b, 2024], convolution mappings such as HoneyBee [Cha et al., 2024] and H-Reducer [Hu et al., 2024], or attention-based mechanisms such as the Perceiver Resampler [Li et al., 2023b, Laurençon et al., 2024, Alayrac et al., 2022]. This approach is more parameter-efficient and computationally lighter than *deep fusion* method However, these connectors lack inductive bias to ensure that the projected features remain within the region spanned by the LLM's pretrained text embeddings. Consequently, the projected visual features may fall outside the distribution the LLM was trained on, leading to noisy or misaligned representations. Moreover, these mappings are typically learned from scratch, making them data-inefficient and less effective under low-resource conditions.

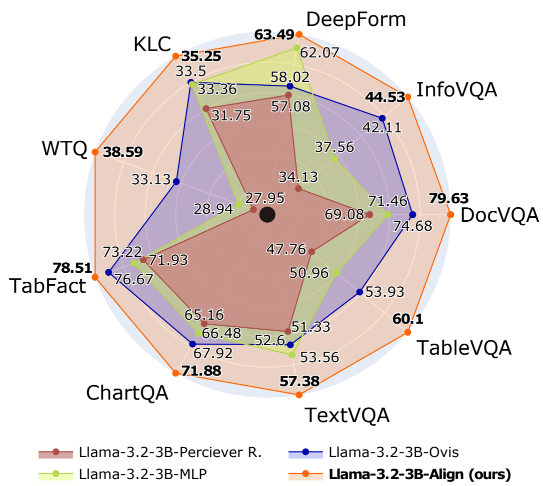


Figure 1: **Performance of Different VLM Connectors.** The proposed **ALIGN** connector outperforms other methods across benchmarks using the same training configuration. Radial distance is proportion of maximal score, truncated at 0.7 (black dot).

Recent methods like Ovis [Lu et al., 2024] attempt to alleviate these issues by introducing separate visual embeddings indexed from the vision encoder outputs and combined together to construct the visual inputs to the LLM. However, this approach significantly increases parameter count due to the massive embedding matrix and requires extensive training to learn a new embedding space without guaranteeing alignment with the LLM's input latent space.

To address these limitations, this paper introduces **ALIGNVLM**, a novel framework that sidesteps direct projection of visual features into the LLM embedding space. Instead, our proposed connector, ALIGN, maps visual features into probability distributions over the LLM's *existing* pretrained vocabulary embeddings, which are then combined into a weighted representation of the text embeddings. By constraining each visual feature as a convex combination of the LLM text embeddings, our approach leverages the linguistic priors already encoded in the LLM's text space. This ensures that the resulting visual features lie within the convex hull of the LLM's embedding space, reducing the risk of noisy or out-of-distribution inputs and improving alignment between modalities. The connector thus enables faster convergence and stronger performance, particularly in low-resource scenarios.

Our experimental results show that ALIGN improves performance on various document understanding tasks, outperforming prior connector methods, with especially large gains in low-data regimes. We summarize our main contributions as follows:

- We propose a novel connector, **ALIGN**, to bridge the representation gap between vision and text modalities.
- We introduce a family of Vision-Language Models, **ALIGNVLM**, that achieves state-of-the-art performance on multimodal document understanding tasks by leveraging ALIGN.
- We conduct extensive experiments demonstrating the robustness and effectiveness of **ALIGN** across different LLM sizes and training data setups.

We release our code and research artifacts at alignvlm.github.io.

2 Related Work

2.1 Vision-Language Models

Over the past few years, Vision-Language Models (VLMs) have achieved remarkable progress, largely due to advances in Large Language Models (LLMs). Initially demonstrating breakthroughs in text understanding and generation [Brown et al., 2020, Raffel et al., 2023, Achiam et al., 2023, Grattafiori et al., 2024, Qwen et al., 2025, Team, 2024], LLMs are now increasingly used to effectively interpret visual inputs [Liu et al., 2023b, Li et al., 2024, Wang et al., 2024, Chen et al., 2024b, Dai et al., 2024, Drouin et al., 2024, Rodriguez et al., 2022]. This progress has enabled real-world applications across diverse domains, particularly in multimodal document understanding for tasks like form reading [Svetlichnaya, 2020], document question answering [Mathew et al., 2021b], and chart question answering [Masry et al., 2022]. VLMs commonly adopt a three-component architecture: a pretrained vision encoder [Zhai et al., 2023, Radford et al., 2021], a LLM, and a connector module. A key challenge for VLMs is effectively aligning visual features with the LLM's semantic space to enable accurate and meaningful multimodal interpretation.

2.2 Vision-Language Alignment for Multimodal Models

Existing vision-language alignment approaches can be classified into *deep fusion* and *shallow fusion*. Deep fusion methods integrate visual and textual features by modifying the LLM's architecture, adding cross-attention and feed-forward layers. For example, Flamingo [Alayrac et al., 2022] employs the Perceiver Resampler, which uses fixed latent embeddings to attend to vision features and fuses them into the LLM via gated cross-attention layers. Similarly, NVLM [Dai et al., 2024] adopts cross-gated attention while replacing the Perceiver Resampler with a simpler MLP. CogVLM [Wang et al., 2023b] extends this approach by incorporating new feed-forward (FFN) and QKV layers for the vision modality within every layer of the LLM. While these methods improve cross-modal alignment, they significantly increase parameter counts and computational overhead, making them less efficient.

On the other hand, shallow fusion methods are more computationally efficient, mapping visual features into the LLM's embedding space without altering its architecture. These methods can be categorized into three main types: *(1) MLP-based mapping*, such as LLaVA [Liu et al., 2023b] and PaliGemma [Beyer et al., 2024], which use multilayer perceptrons (MLP) to project visual features but often produce misaligned or noisy features due to a lack of constraints and inductive bias [Rodriguez et al., 2024b]; *(2) cross-attention mechanisms*, BLIP-2 [Li et al., 2023b] uses Q-Former, which utilizes a fixed set of latent embeddings to cross-attend to visual features, but that may still produce noisy or OOD visual features; *(3) convolution-based mechanisms*, such as HoneyBee [Cha et al., 2024] and H-Reducer [Hu et al., 2024], which leverage convolutional or ResNet [He et al., 2015] layers to preserve spatial locality while reducing dimensionality; and *(4) visual embeddings*, such as those introduced by Ovis [Lu et al., 2024], which use embeddings indexed by the vision encoder's outputs to produce the visual inputs. While this regularizes feature mapping, it adds substantial parameter overhead and creates a new vision embedding space, risking misalignment with the LLM's text embedding space. Encoder-free VLMs, like Fuyu-8B [1] and EVE [Diao et al., 2024], eliminate dedicated vision encoders but show degraded performance [Beyer et al., 2024].

In contrast, ALIGNVLM maps visual features from the vision encoder into probability distributions over the LLM's text embeddings, using them to compute a convex combination. By leveraging the linguistic priors encoded in the LLM's vocabulary, ALIGNVLM ensures that visual features remain within the convex hull of the text embedding. This design mitigates noisy or out-of-distribution projections and achieves stronger multimodal alignment, particularly in tasks that require joint modalities representation like multimodal document understanding and in low-resource settings.

3 Methodology

3.1 Model Architecture

The overall model architecture, shown in Figure 2, consists of three main components:

[1] https://www.adept.ai/blog/fuyu-8b

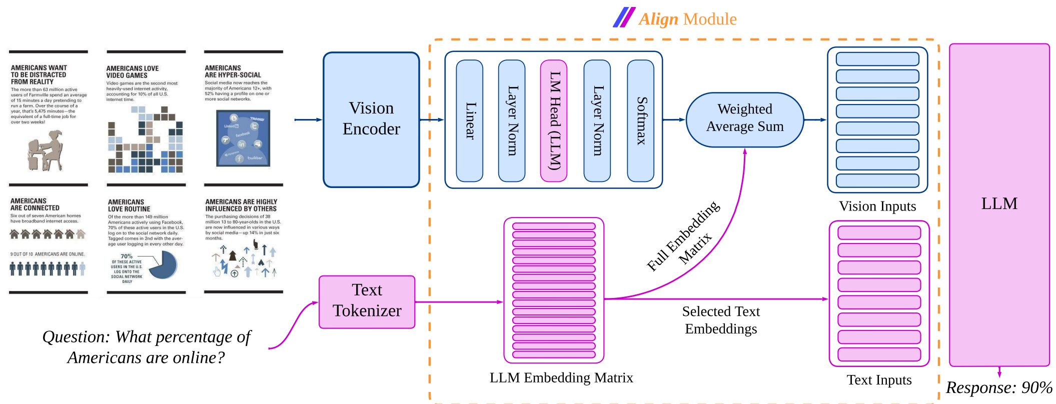


Figure 2: **ALIGNVLM Model Architecture.** The vision encoder extracts image features, which are processed to produce probabilities over the LLM embeddings. A weighted average combines these probabilities with embeddings to generate vision input vectors. Text inputs are tokenized, and the corresponding embeddings are selected from the embedding matrix, which is then used as input to the LLM. We display the vision layers in blue, and the text layers in purple.

(1) Vision Encoder. To handle high-resolution images of different aspect ratios, we divide each input image into multiple tiles according to one of the predefined aspect ratios (e.g., 1:1, 1:2, . . . , 9:1) chosen via a coverage ratio [Lu et al., 2024, Chen et al., 2024a]. Due to limited computational resources, we set the maximum number of tiles to 9. Each tile is further partitioned into 14×14 patches, projected into vectors, and processed by a SigLip-400M vision encoder [Zhai et al., 2023] to extract contextual visual features.

Each tile $t \in \{1, \ldots, T\}$ is divided into N_t patches

$$\mathbf{P}_t = \{\mathbf{p}_{t,1}, \ldots, \mathbf{p}_{t,N_t}\},$$

where $\mathbf{p}_{t,i}$ is the i-th patch of tile t. The vision encoder maps these patches to a set of visual feature vectors

$$\mathbf{F}_t = \text{VisionEncoder}(\mathbf{P}_t), \quad \mathbf{F}_t = \{\mathbf{f}_{t,1}, \ldots, \mathbf{f}_{t,N_t}\}, \quad \mathbf{f}_{t,i} \in \mathbb{R}^d.$$

Finally, we concatenate the feature sets across all tiles into a single output

$$\mathbf{F} = \text{concat}\Big(\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_T\Big).$$

(2) ALIGN Module. This module aligns the visual features with the LLM. A linear layer $\mathbf{W}_1 \in \mathbb{R}^{D \times d}$ first projects the visual features $\mathbf{F} \in \mathbb{R}^{T \cdot N_t \times d}$ to the LLM's token embedding space: one $\mathbb{R}^D$ vector per token. A second linear layer $\mathbf{W}_2 \in \mathbb{R}^{V \times D}$ (initialized from the LLM's language-model head) followed by a softmax, produces a probability simplex $\mathbf{P}_{\text{vocab}}$ over the LLM's vocabulary (V tokens)

$$\mathbf{P}_{\text{vocab}} = \text{softmax}(\text{LayerNorm}(\mathbf{W}_2\,\text{LayerNorm}(\mathbf{W}_1\mathbf{F}))) \tag{1}$$

We then use the LLM text embeddings $\mathbf{E}_{\text{text}} \in \mathbb{R}^{V \times D}$ to compute a weighted sum

$$\mathbf{F}'_{\text{align}} = \mathbf{P}^{\top}_{\text{vocab}}\mathbf{E}_{\text{text}}. \tag{2}$$

Finally, we concatenate $\mathbf{F}'_{\text{align}}$ with the tokenized text embeddings to form the LLM input

$$\mathbf{H}_{\text{input}} = \text{concat}\big(\mathbf{F}'_{\text{align}}, \mathbf{E}_{\text{text}}(\mathbf{x})\big),$$

where $\mathbf{E}_{\text{text}}(\mathbf{x})$ is obtained by tokenizing the input text $\mathbf{x} = (x_1, \ldots, x_M)$ and selecting the corresponding embeddings from $\mathbf{E}_{\text{text}}$ such that

$$\mathbf{E}_{\text{text}}(\mathbf{x}) = \big[\mathbf{E}_{\text{text}}(x_1), \ldots, \mathbf{E}_{\text{text}}(x_M)\big]. \tag{3}$$

(3) Large Language Model. We feed the concatenated vision and text vectors, $\mathbf{H}_{\text{input}}$, into the LLM, which then generates output text auto-regressively. To demonstrate the effectiveness of our alignment technique, we experiment with the Llama 3.1 model family [Grattafiori et al., 2024]. These models offer state-of-the-art performance and permissive licenses, making them suitable for commercial applications. In particular, we utilize Llama 3.2-1B, Llama 3.2-3B, and Llama 3.1-8B.

3.2 Motivation and relation with existing methods

By construction, each $\mathbb{R}^D$ representation in $\mathbf{F}'_{\text{align}}$ is constrained to the convex hull of the points $\mathbb{E}_{\text{text}}$, thus concentrating the visual features in the part of latent space that the LLM can effectively interpret. Moreover, we argue that our initialization of $\mathbf{W}_2$ to the language model head is an inductive bias toward *recycling* some of the semantics of these text tokens into visual tokens. This contrasts with past methods that have been proposed to adapt the vision encoder outputs $\mathbf{F} \in \mathbb{R}^{T \times N_t \times d}$ to an $\mathbf{F}' \in \mathbb{R}^{T \times N_t \times D}$ to be fed to the LLM. Here, we consider two examples in more detail, highlighting these contrasts.

(1) MLP Connector Liu et al. [2023b] applies a linear projection with parameters $\mathbf{W}_{\text{MLP}} \in \mathbb{R}^{D \times d}$ and $\mathbf{b}_{\text{MLP}} \in \mathbb{R}^{D}$, followed by an activation function σ (e.g., ReLU)

$$\mathbf{F}'_{\text{MLP}} = \sigma(\mathbf{W}_{\text{MLP}}\mathbf{F} + \mathbf{b}_{\text{MLP}}).$$

These parameters are all learned from scratch, without any bias aligning them to text embeddings.

(2) Visual Embedding Table Lu et al. [2024] introduces an entire new set of visual embeddings $\mathbf{E}_{\text{VET}} \in \mathbb{R}^{K \times D}$ which, together with the weights $\mathbf{W}_{\text{VET}} \in \mathbb{R}^{K \times d}$, specifies

$$\mathbf{F}'_{\text{VET}} = \mathsf{softmax}(\mathbf{W}_{\text{VET}}\mathbf{F})^{\top}\mathbf{E}_{\text{VET}}.$$

When $D < d$, our $\mathbf{W}_2\mathbf{W}_1$ amounts to a low-rank version of $\mathbf{W}_{\text{VET}}$. There is thus much more to learn to obtain $\mathbf{F}'_{\text{VET}}$, and there is again no explicit pressure to align it with the text embeddings.

3.3 Training Datasets & Stages

We train our model in three stages:

Stage 1. This stage focuses on training the ALIGN Module to map visual features to the LLM's text embeddings effectively. We use the CC-12M dataset Changpinyo et al. [2021], a large-scale web dataset commonly used for VLM pretraining Liu et al. [2023b], which contains 12M image-text pairs. However, due to broken or unavailable links, we retrieved 8.1M pairs. This dataset facilitates the alignment of visual features with the text embedding space of the LLM. During this stage, we train the full model, as this approach improves performance and stabilizes the ALIGN Module training.

Stage 2. The goal is to enhance the model's document understanding capabilities, such as OCR, document structure comprehension, in-depth reasoning, and instruction-following. We leverage the BigDocs-7.5M dataset Rodriguez et al. [2024a], a curated collection of license-permissive datasets for multimodal document understanding. This dataset aligns with the Accountability, Responsibility, and Transparency (ART) principles Bommasani et al. [2023], Vogus and Llansóe [2021], ensuring compliance for commercial applications. As in Stage 1, we train the full model during this stage.

Stage 3. To enhance the model's instruction-tuning capabilities, particularly for downstream tasks like question answering, we further train it on the DocDownstream Rodriguez et al. [2024a], Hu et al. [2024] instruction tuning dataset. In this stage, the vision encoder is frozen, focusing training exclusively on the LLM and ALIGN module.

4 Experimental Setup

Setup. We conduct all experiments using 8 nodes of H100 GPUs, totaling 64 GPUs. For model training, we leverage the MS-Swift framework [Zhao et al., 2024] for its flexibility. Additionally, we utilize the DeepSpeed framework [Aminabadi et al., 2022], specifically the ZeRO-3 configuration, to optimize efficient parallel training across multiple nodes. Detailed hyperparameters are outlined in Appendix A.1.

Table 1: **Main Results on General Document Benchmarks.** We compare ALIGNVLM (ours) with state-of-the-art (SOTA) open and closed-source instructed models, and with base models that we trained using the process described in Section 3.3. ALIGNVLM models outperform all Base VLM models trained in the same data regime. Our models also perform competitively across document benchmarks even compared with SOTA models, in which the data regime is more targeted and optimized. Color coding for comparison: closed-source models, open-source models below 7B parameters, open-source models between 7-12B parameters.

Model	DocVQA VAL	InfoVQA VAL	DeepForm TEST	KLC TEST	WTQ TEST	TabFact TEST	ChartQA TEST	TextVQA VAL	TableVQA TEST	Avg. Score
Closed-Source VLMs (*Opaque Training Data*)										
Claude-3.5 Sonnet	88.48	59.05	31.41	24.82	47.13	53.48	51.84	**71.42**	**81.27**	56.54
GeminiPro-1.5	91.23	**73.94**	32.16	24.07	**50.29**	71.22	34.68	68.16	80.43	58.46
GPT-4o 20240806	**92.80**	66.37	**38.39**	**29.92**	46.63	**81.10**	**85.70**	70.46	72.87	**64.91**
Open-Source Instruct VLMs (*Semi-Opaque Training Data*)										
Janus-**1.3B** [Wu et al., 2024a]	30.15	17.09	0.62	15.06	9.30	51.34	57.20	51.97	18.67	27.93
Qwen2-VL-**2B** [Wang et al., 2024]	89.16	64.11	32.38	25.18	38.20	57.21	73.40	79.90	43.07	55.84
Qwen2.5-VL-**3B** [Wang et al., 2024]	**93.00**	**75.83**	32.84	24.82	**53.46**	**71.16**	**83.91**	79.29	**71.66**	**65.10**
InternVL-2.5-**2B** [Chen et al., 2024b]	87.70	61.85	13.14	16.58	36.33	57.26	74.96	76.85	42.20	51.87
InternVL-3-**2B** [Zhu et al., 2025]	87.33	66.99	**37.90**	**29.79**	39.44	59.91	75.32	78.69	43.46	57.64
DeepSeek-VL2-Tiny-**3.4B** [Wu et al., 2024b]	88.57	63.88	25.11	19.04	35.07	52.15	80.92	**80.48**	56.30	55.72
Phi3.5-Vision-**4B** [Abdin et al., 2024]	86.00	56.20	10.47	7.49	17.18	30.43	82.16	73.12	70.70	48.19
Qwen2-VL-**7B** [Wang et al., 2024]	93.83	76.12	34.55	23.37	52.52	74.68	83.16	84.48	53.97	64.08
Qwen2.5-VL-**7B** [Bai et al., 2025]	**94.88**	**82.49**	42.21	24.26	**61.96**	78.56	**86.00**	**85.35**	**76.10**	**70.20**
LLaVA-NeXT-**7B** [Xu et al., 2024]	63.51	30.90	1.30	5.35	20.06	52.83	52.12	65.10	32.87	36.00
DocOwl1.5-**8B** [Hu et al., 2024]	80.73	49.94	**68.84**	**37.99**	38.87	**79.67**	68.56	68.91	52.60	60.68
InternVL-2.5-**8B** [Chen et al., 2024b]	91.98	75.36	34.55	22.31	50.33	74.75	82.84	79.00	52.10	62.58
InternVL-3-**8B** [Zhu et al., 2025]	91.99	73.90	51.24	36.41	53.60	72.27	85.60	82.41	53.26	66.74
Fuyu-**8B** [Bavishi et al., 2023]	48.97	23.09	4.78	6.63	14.55	47.91	44.36	46.02	15.49	22.97
Ovis-1.6-Gemma2-**9B** [Lu et al., 2024]	88.84	73.97	45.16	23.91	50.72	76.66	81.40	77.73	48.33	62.96
Llama3.2-**11B** [Grattafiori et al., 2024]	82.71	36.62	1.78	3.47	23.03	58.33	23.80	54.28	22.40	34.04
Pixtral-**12B** [Agrawal et al., 2024]	87.67	49.45	27.37	24.07	45.18	73.53	71.80	76.09	67.13	58.03
Document Understanding Instructed Models (*Instruction Tuned on BigDocs-7.5M + DocDownStream [Rodriguez et al., 2024a, Hu et al., 2024]*)										
Qwen2-VL-**2B** (base+) [Wang et al., 2024]	57.23	31.88	49.31	34.39	31.61	64.75	68.60	**61.01**	47.53	49.59
ALIGNVLM-Llama-3.2-**1B** (ours)	72.42	38.16	60.47	33.71	28.66	71.31	65.44	48.81	50.29	52.14
ALIGNVLM-Llama-3.2-**3B** (ours)	**79.63**	**44.53**	**63.49**	**35.25**	**38.59**	**78.51**	**71.88**	57.38	**60.10**	**58.81**
DocOwl1.5-**8B** (base+) [Hu et al., 2024]	78.70	47.62	64.39	36.93	35.69	72.65	65.80	67.30	49.03	57.56
Llama3.2-**11B** (base+) [Grattafiori et al., 2024]	78.99	44.27	**67.05**	**37.22**	40.18	78.04	71.40	**68.46**	56.73	60.26
ALIGNVLM-Llama-3.1-**8B** (ours)	**81.18**	**53.75**	63.25	35.50	**45.31**	**83.04**	**75.00**	64.60	**64.33**	**62.88**

Baselines. Our work focuses on architectural innovations, so we ensure that all baselines are trained on the same datasets. To enable fair comparisons, we evaluate our models against a set of **Base VLMs** fine-tuned on the same instruction-tuning tasks (Stages 2 and 3) as our models, using the BigDocs-7.5M and BigDocs-DocDownstream datasets. This approach ensures consistent training data, avoiding biases introduced by the **Instruct** versions of VLMs, which are often trained on undisclosed instruction-tuning datasets. Due to the scarcity of recently released publicly available Base VLMs, we primarily compare our model against the following Base VLMs of varying sizes: Qwen2-VL-2B [Wang et al., 2024], DocOwl1.5-8B [Hu et al., 2024], and LLama 3.2-11B [Grattafiori et al., 2024].

For additional context, we also include results from the Instruct versions of recent VLMs of different sizes: Phi3.5-Vision-4B [Abdin et al., 2024], Qwen2-VL-2B and 7B [Wang et al., 2024], Qwen2.5-VL-7B [Qwen et al., 2025], LLaVA-NeXT-7B [Liu et al., 2024], InternVL2.5-2B and 8B [Chen et al., 2024b], InternVL3-2B and 8B [Zhu et al., 2025], Janus-1.3B [Wu et al., 2024a], DeepSeek-VL2-Tiny [Wu et al., 2024b], Ovis1.6-Gemma-9B [Lu et al., 2024], Llama3.2-11B [Grattafiori et al., 2024], DocOwl1.5-8B [Hu et al., 2024], and Pixtral-12B [Agrawal et al., 2024].

Evaluation Benchmarks. We evaluate our models on a diverse range of document understanding benchmarks that assess the model's capabilities in OCR, chart reasoning, table processing, or form comprehension. In particular, we employ the VLMEvalKit [Duan et al., 2024] framework and report the results on the following popular benchmarks: DocVQA [Mathew et al., 2021b], InfoVQA [Mathew et al., 2021a], DeepForm [Svetlichnaya, 2020], KLC [Stanisławek et al., 2021], WTQ [Pasupat and Liang, 2015], TabFact [Chen et al., 2020], ChartQA [Masry et al., 2022], TextVQA [Singh et al., 2019], and TableVQA [Kim et al., 2024].

Table 2: **Impact of Connector Designs on VLM Performance:** We present the results of experiments evaluating different connector designs for conditioning LLMs on visual features. Our proposed **ALIGN** connector is compared against a basic Multi-Layer Perceptron (**MLP**), the **Perceiver Resampler**, and **Ovis**. The results demonstrate that ALIGN consistently outperforms these alternatives across all benchmarks.

Model	DocVQA (VAL)	InfoVQA (VAL)	DeepForm (TEST)	KLC (TEST)	WTQ (TEST)	TabFact (TEST)	ChartQA (TEST)	TextVQA (VAL)	TableVQA (TEST)	Avg. Score
Llama-3.2-3B-**MLP**	71.46	37.56	62.07	33.36	28.94	73.22	66.48	53.56	50.96	53.06
Llama-3.2-3B-**Perciever R.**	69.08	34.13	57.08	31.75	27.95	71.93	65.16	51.33	47.76	50.68
Llama-3.2-3B-**Ovis**	74.68	42.11	58.02	33.50	33.13	76.67	67.92	52.60	53.93	54.72
Llama-3.2-3B-**ALIGN** (ours)	**79.63**	**44.53**	**63.49**	**35.25**	**38.59**	**78.51**	**71.88**	**57.38**	**60.10**	**58.81**

5 Results

5.1 Main Results

Table 1 presents the performance of ALIGNVLM compared to state-of-the-art (SOTA) open- and closed-source instructed models, as well as baseline Base VLMs fine-tuned in the same instruction-tuning setup. The results demonstrate that ALIGNVLM consistently outperforms all Base VLMs within the same size category and achieves competitive performance against SOTA Instruct VLMs despite being trained on a more limited data regime. Below, we provide a detailed analysis.

ALIGNVLM vs. Base VLMs. Our ALIGNVLM models, based on Llama 3.2-1B and Llama 3.2-3B, significantly outperform the corresponding Base VLM, Qwen2-VL-2B, by up to *9.22%*. Notably, ALIGNVLM-Llama-3.2-3B surpasses DocOwl1.5-8B, which has 4B more parameters, demonstrating the effectiveness of ALIGN in enhancing multimodal capabilities compared to traditional *shallow fusion* methods (e.g., MLPs). Furthermore, our 8B model achieves a *2.62%* improvement over Llama3.2-11B despite sharing the same Base LLM, Llama3.1-8B. Since all models in this comparison were trained on the same instruction-tuning setup, this experiment provides a controlled evaluation, isolating the impact of architectural differences rather than dataset biases. Consequently, these results suggest that ALIGNVLM outperforms VLMs with shallow fusion techniques and surpasses parameter-heavy *deep fusion* VLMs, such as Llama3.2-11B, while maintaining a more efficient architecture.

ALIGNVLM vs. Instruct VLMs. Even as open-source Instruct models are trained on significantly larger, often undisclosed instruction-tuning datasets, ALIGNVLM achieves competitive performance. For example, ALIGNVLM-Llama-3.2-3B (*58.81%*) outperforms other strong instruction-tuned VLMs in its size class, such as Qwen2-VL-2B and InternVL-3-2B, by considerable margins (*2.97%* and *1.17%*, respectively). While it falls slightly behind Qwen2.5-VL-3B, a direct comparison is not entirely fair, as the latter was trained on a proprietary instruction-tuning dataset.

Additionally, our 8B model outperforms significantly larger models such as Llama 3.2-11B and PixTral-12B by substantial margins. It also surpasses InternVL-2.5-8B and performs competitively with Qwen2.5-VL-7B, though a direct comparison may not be entirely fair since Qwen2.5-VL-7B was trained on an undisclosed instruction-tuning dataset. Finally, ALIGNVLM also exhibits comparable performance to closed-source models like GeminiPro-1.5 and GPT4o.

Overall, these results validate the effectiveness of ALIGN and establish ALIGNVLM as a state-of-the-art model for multimodal document understanding.

5.2 Impact of Connector Designs on VLM Performance

5.2.1 High-Resource Training Regime

To assess the effectiveness of our ALIGN module, we compare it against three different and widely used *shallow fusion* VLM connectors: MLP, Perceiver Resampler, and Ovis. These experiments were carefully conducted under precisely identical training conditions (datasets, hyperparameters, training stages) as outlined in Appendix A.1, ensuring a fair and rigorous comparison. The results in Table 2 show that ALIGN consistently outperforms all alternatives, demonstrating its superiority

Table 3: **Connector Performance under a Low-Resource Training Regime:** We evaluate the effectiveness of more shallow-fusion connectors when trained on limited data. The ALIGN connector achieves the highest performance, with notably larger gains on document understanding tasks, demonstrating its data efficiency and strong inductive bias.

Model	Document Understanding Tasks					General Vision Tasks					
	DocVQA	InfoVQA	ChartQA	TextVQA	Avg.	MMMU	SeedBench	MMVet	POPE	GQA	Avg.
Llama-3.2-3B-MLP	42.11	19.93	48.44	51.97	40.61	33.33	58.54	31.14	87.35	57.62	53.59
Llama-3.2-3B-Perceiver	32.18	18.10	40.00	44.31	33.64	35.22	**63.70**	26.19	84.92	55.86	53.17
Llama-3.2-3B-Ovis	57.73	26.39	54.52	55.60	48.56	31.89	60.97	30.41	88.26	56.23	53.55
Llama-3.2-3B-Hreducer	34.59	17.57	45.64	47.13	36.23	35.00	61.82	28.39	87.48	58.24	54.18
Llama-3.2-3B-HoneyBee	55.86	19.36	55.32	58.13	47.16	32.11	61.18	34.31	**89.28**	54.79	54.33
Llama-3.2-3B-ALIGN **(ours)**	**71.43**	**30.50**	**69.72**	**65.63**	**59.32**	**35.33**	63.27	**35.32**	88.85	**61.67**	**56.88**

both in aligning visual and textual modalities in multimodal document understanding. MLP and Perceiver Resampler achieve the lowest performance, 53.06% and 50.68%, respectively, due to their direct feature projection, which lacks an explicit mechanism to align visual features with the LLM's text space, leading to misalignment. Ovis introduces a separate visual embedding table, but this additional complexity does not significantly improve alignment, yielding only 54.72% accuracy. In contrast, ALIGN ensures that visual features remain within the convex hull of the LLM's text latent space, leveraging the linguistic priors of the LLM to enhance alignment and mitigate noisy embeddings. This design leads to the highest performance (58.81%), establishing ALIGN as the most effective connector for integrating vision and language in multimodal document understanding. We provide some example outputs of the Llama-3.2-3B models with different connector designs in Appendix A.4. Furthermore, we include an analysis of the runtime efficiency and memory usage of different connectors in Appendix A.2.

5.2.2 Low-Resource Training Regime

The previous section focused on large-scale training setups involving millions of data samples (BigDocs-7.5M), which require significant compute resources and limit the number of baselines that we were able to compare against. Here, we examine whether ALIGN remains effective in a *low-resource* setting.

We conduct additional experiments using SigLIP-400M as the vision encoder and Llama-3.2-3B as the language model, fine-tuned on the LLaVA-NeXT dataset Liu et al. [2024], which contains *779K* samples. We follow the official LLaVA-NeXT configuration for both training stages. *(i) Pretraining:* the model is trained on the LLaVA-558K image–caption dataset Liu et al. [2024], freezing both the LLM and vision encoder while fine-tuning the connector (learning rate = 1e-3, batch size = 32, 1 epoch on 8 × H100 GPUs). To handle high-resolution document images, we adopt the "anyres_max_9" strategy with grid weaving from 1×1 to 6×6, supporting resolutions up to 2304×2304 with 729 tokens per grid; *(ii) Instruction tuning:* the model is further fine-tuned on the LLaVA-NeXT-779K instruction dataset with learning rates of 1e-5 for the LLM and connector, 2e-6 for the vision encoder, batch size = 8, for 1 epoch.

This lightweight setup allows direct comparison across more connector architectures including MLP Liu et al. [2023a], Perceiver Resampler, Ovis Lu et al. [2024], H-Reducer (1×4) Hu et al. [2024], and HoneyBee (C-Abstractor) Cha et al. [2024], all trained under identical conditions for fairness. Since the LLaVA-Next dataset is general-purpose and not exclusively document-focused like BigDocs-7.5M [Rodriguez et al., 2024a], it allows us to evaluate whether the ALIGN connector generalizes beyond document understanding to broader visual reasoning. Accordingly, we assess all models on a comprehensive suite of benchmarks spanning both document understanding and general vision–language tasks. The document understanding benchmarks include DocVQA Mathew et al. [2021b], InfoVQA Mathew et al. [2021a], ChartQA Masry et al. [2022], and TextVQA Singh et al. [2019]. For general vision–language evaluation, we report results on MMMU-dev Yue et al. [2024], SeedBench Li et al. [2023a], and MMVet Yu et al. [2024], Pope [Li et al., 2023c], and GQA [Hudson and Manning, 2019].

As summarized in Table 3, ALIGN consistently outperforms other connectors under this low-data regime, with stronger gains on document understanding tasks. The wider performance margin

Table 4: Performance comparison when evaluating ALIGN with the full text embedding vocabulary (128K) versus the reduced subset of 3.4K high-probability embeddings. The results show negligible performance degradation, indicating that ALIGN relies primarily on a small subset of embeddings.

Model	DocVQA VAL	InfoVQA VAL	DeepForm TEST	KLC TEST	WTQ TEST	TabFact TEST	ChartQA TEST	TextVQA VAL	TableVQA TEST	Avg. Score
Llama-3.2-3B-ALIGN **(Full Embeddings)**	79.63	44.53	63.49	35.25	38.59	78.51	71.88	57.38	60.10	58.81
Llama-3.2-3B-ALIGN **(3.4K Embeddings)**	79.40	44.13	63.64	35.02	38.26	78.83	71.72	57.48	59.80	58.69

between ALIGN and others connectors under limited data (Table 3) compared to the high-resource setting (Table 2) underscores the benefit of its inductive bias. By grounding visual features within the LLM's text embedding space, ALIGN learns more efficiently from fewer samples, unlike direct-projection connectors that rely heavily on large datasets. This makes ALIGN especially valuable for resource-constrained environments such as academic labs or small-scale industrial research setups, where both data and compute are limited.

5.3 Probability Distribution over Text Tokens Analysis

To better understand the behavior of ALIGN, we examine the probability distribution, $\mathbf{P}_{\text{vocab}}$ in Eq (1), over the LLM's text vocabulary generated from visual features. Specifically, we process 100 document images through the vision encoder and ALIGN, then average the resulting probability distributions across all image patches. The final distribution is shown in Figure 3. As illustrated, the distribution is *dense* (rather than sparse), with the highest probability assigned to a single token being *0.0118*. This can be explained by the vision feature space being continuous and of much higher cardinality than the discrete text space. Indeed, while the LLM has 128K distinct vocabulary tokens, an image patch (e.g., 14×14 pixels) contains continuous, high-dimensional information that cannot be effectively mapped to a single or a few discrete tokens.

We conducted a deeper analysis of the token probability distributions produced by the ALIGN connector. Our observations show that ALIGN consistently assigns high probabilities to approximately 3.4K tokens from the entire vocabulary, while the remaining tokens receive negligible probabilities (below 10^{-6}). To better understand this behavior, we applied Principal Component Analysis (PCA) to reduce the dimensionality of the embeddings and visualized them in a two-dimensional space, as shown in Figure 4. The visualization reveals that these 3.4K tokens densely and comprehensively span the latent space of the LLM's text embeddings. To validate this finding, we conducted additional evaluation experiments in which we retained only these 3.4K high-probability embeddings in the ALIGN connector, entirely removing the rest during evaluation. As shown in Table 4, the performance difference compared to using the full embedding set (128K) was negligible. This confirms that ALIGN effectively leverages and combines a compact subset of embeddings to map visual features into semantically meaningful regions within the LLM's latent text space. Moreover, this suggests that ALIGN can be further optimized through targeted embedding pruning to improve computational efficiency without sacrificing performance.

5.4 Robustness to Noise Analysis

To evaluate the robustness of our **ALIGN** connector to noisy visual features, we conduct an experiment where random Gaussian noise is added to the visual features produced by the vision encoder before passing them into the connector. Specifically, given the visual features $\mathbf{F} \in \mathbb{R}^{N \times d}$ output by the vision encoder (where N is the number of feature vectors and d is their dimensionality), we perturbed them as

$$\widetilde{\mathbf{F}} = \mathbf{F} + \mathbf{N}, \quad \mathbf{N} \sim \mathcal{N}(0, \sigma = 3).$$

Table 5: **Robustness to Noise.** Comparison of Avg. Scores with and without Gaussian noise ($\sigma = 3$), including performance drop ( ).

Model	Without Noise	With Noise	Drop ( )
Llama-3.2-3B-MLP	53.06	27.52	# 25.54
Llama-3.2-3B-ALIGN (ours)	**58.81**	**57.14**	# **1.67**

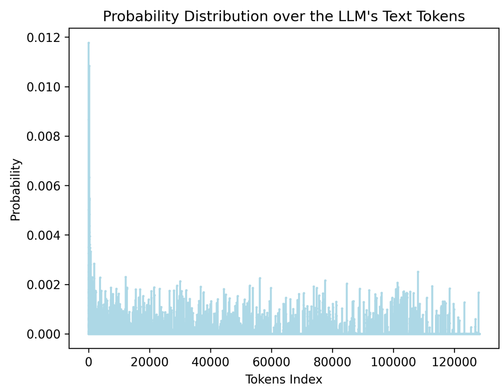


Figure 3: **Probability distribution over LLM tokens**, highlighting dense probabilities for whitespace tokens.

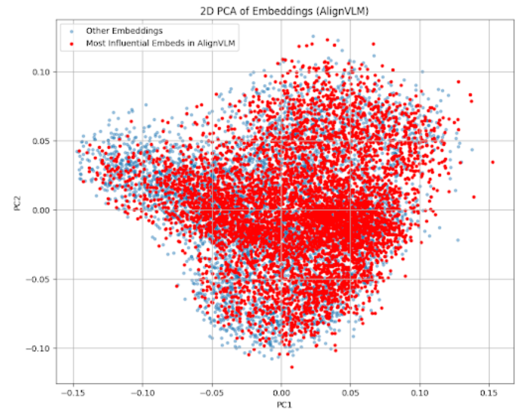


Figure 4: **PCA of ALIGN Embeddings:** The principal components of the most influential embeddings in the Align Connector span most of the feature space represented by all embeddings.

As shown in Table 5, our **ALIGN** connector demonstrates high robustness to noise, with only a *1.67%* average drop in performance. In contrast, the widely adopted MLP connector suffers a significant performance degradation of *25.54%*, highlighting its vulnerability to noisy inputs. Furthermore, we measured the average cosine distance between the original and noise-perturbed visual embeddings using both the ALIGN and MLP connectors. ALIGN showed significantly lower distances (0.0036) than MLP (0.3938), further validating its robustness to noise. These empirical results support our hypothesis that leveraging the knowledge encoded in the LLM's text embeddings and constraining the visual features within the convex hull of the text latent space act as a regularization mechanism, reducing the model's sensitivity to noisy visual features.

6 Conclusion

We introduce ALIGN, a novel connector designed to align vision and language latent spaces in vision-language models (VLMs), specifically enhancing multimodal document understanding. By improving cross-modal alignment and minimizing noisy embeddings, our models, ALIGNVLM, which leverage ALIGN, achieve state-of-the-art performance across diverse document understanding tasks. This includes outperforming base VLMs trained on the same datasets and achieving competitive performance with open-source instruct models trained on undisclosed data. Extensive experiments and ablations validate the robustness and effectiveness of ALIGN compared to existing connector designs, establishing it as a significant contribution to vision-language modeling. Future work will explore training on more diverse instruction-tuning datasets to generalize to broader domains.

References

M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, Q. Cai, V. Chaudhary, D. Chen, D. Chen, W. Chen, Y.-C. Chen, Y.-L. Chen, H. Cheng, P. Chopra, X. Dai, M. Dixon, R. Eldan, V. Fragoso, J. Gao, M. Gao, M. Gao, A. Garg, A. D. Giorno, A. Goswami, S. Gunasekar, E. Haider, J. Hao, R. J. Hewett, W. Hu, J. Huynh, D. Iter, S. A. Jacobs, M. Javaheripi, X. Jin, N. Karampatziakis, P. Kauffmann, M. Khademi, D. Kim, Y. J. Kim, L. Kurilenko, J. R. Lee, Y. T. Lee, Y. Li, Y. Li, C. Liang, L. Liden, X. Lin, Z. Lin, C. Liu, L. Liu, M. Liu, W. Liu, X. Liu, C. Luo, P. Madan, A. Mahmoudzadeh, D. Majercak, M. Mazzola, C. C. T. Mendes, A. Mitra, H. Modi, A. Nguyen, B. Norick, B. Patra, D. Perez-Becker, T. Portet, R. Pryzant, H. Qin, M. Radmilac, L. Ren, G. de Rosa, C. Rosset, S. Roy, O. Ruwase, O. Saarikivi, A. Saied, A. Salim, M. Santacroce, S. Shah, N. Shang, H. Sharma, Y. Shen, S. Shukla, X. Song, M. Tanaka, A. Tupini, P. Vaddamanu, C. Wang, G. Wang, L. Wang, S. Wang, X. Wang, Y. Wang, R. Ward, W. Wen, P. Witte, H. Wu, X. Wu, M. Wyatt, B. Xiao, C. Xu, J. Xu, W. Xu, J. Xue, S. Yadav, F. Yang, J. Yang, Y. Yang, Z. Yang, D. Yu, L. Yuan, C. Zhang, C. Zhang, J. Zhang, L. L. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, and X. Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL `https://arxiv.org/abs/2404.14219`.

J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

P. Agrawal, S. Antoniak, E. B. Hanna, B. Bout, D. Chaplot, J. Chudnovsky, D. Costa, B. D. Monicault, S. Garg, T. Gervet, S. Ghosh, A. Héliou, P. Jacob, A. Q. Jiang, K. Khandelwal, T. Lacroix, G. Lample, D. L. Casas, T. Lavril, T. L. Scao, A. Lo, W. Marshall, L. Martin, A. Mensch, P. Muddireddy, V. Nemychnikova, M. Pellat, P. V. Platen, N. Raghuraman, B. Rozière, A. Sablayrolles, L. Saulnier, R. Sauvestre, W. Shang, R. Soletskyi, L. Stewart, P. Stock, J. Studnia, S. Subramanian, S. Vaze, T. Wang, and S. Yang. Pixtral 12b, 2024. URL `https://arxiv.org/abs/2410.07073`.

J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL `https://arxiv.org/abs/2204.14198`.

R. Y. Aminabadi, S. Rajbhandari, M. Zhang, A. A. Awan, C. Li, D. Li, E. Zheng, J. Rasley, S. Smith, O. Ruwase, and Y. He. Deepspeed inference: Enabling efficient inference of transformer models at unprecedented scale, 2022. URL `https://arxiv.org/abs/2207.00032`.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024.

S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report, 2025. URL `https://arxiv.org/abs/2502.13923`.

R. Bavishi, E. Elsen, C. Hawthorne, M. Nye, A. Odena, A. Somani, and S. Taşırlar. Introducing our multimodal models, 2023. URL `https://www.adept.ai/blog/fuyu-8b`.

L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. Salz, M. Neumann, I. Alabdulmohsin, M. Tschannen, E. Bugliarello, T. Unterthiner, D. Keysers, S. Koppula, F. Liu, A. Grycner, A. Gritsenko, N. Houlsby, M. Kumar, K. Rong, J. Eisenschlos, R. Kabra, M. Bauer, M. Bošnjak, X. Chen, M. Minderer, P. Voigtlaender, I. Bica, I. Balazevic, J. Puigcerver, P. Papalampidi, O. Henaff, X. Xiong, R. Soricut, J. Harmsen, and X. Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL `https://arxiv.org/abs/2407.07726`.

R. Bommasani, K. Klyman, S. Longpre, S. Kapoor, N. Maslej, B. Xiong, D. Zhang, and P. Liang. The foundation model transparency index, 2023. URL `https://arxiv.org/abs/2310.12941`.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

J. Cha, W. Kang, J. Mun, and B. Roh. Honeybee: Locality-enhanced projector for multimodal llm, 2024. URL `https://arxiv.org/abs/2312.06742`.

S. Changpinyo, P. Sharma, N. Ding, and R. Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts, 2021. URL `https://arxiv.org/abs/2102.08981`.

W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, and W. Y. Wang. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference Learning Representations*, 2020.

Z. Chen, W. Wang, H. Tian, S. Ye, Z. Gao, E. Cui, W. Tong, K. Hu, J. Luo, Z. Ma, J. Ma, J. Wang, X. Dong, H. Yan, H. Guo, C. He, B. Shi, Z. Jin, C. Xu, B. Wang, X. Wei, W. Li, W. Zhang, B. Zhang, P. Cai, L. Wen, X. Yan, M. Dou, L. Lu, X. Zhu, T. Lu, D. Lin, Y. Qiao, J. Dai, and W. Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites, 2024a. URL `https://arxiv.org/abs/2404.16821`.

Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024b.

W. Dai, N. Lee, B. Wang, Z. Yang, Z. Liu, J. Barker, T. Rintamaki, M. Shoeybi, B. Catanzaro, and W. Ping. Nvlm: Open frontier-class multimodal llms. *arXiv preprint arXiv: 2409.11402*, 2024.

H. Diao, Y. Cui, X. Li, Y. Wang, H. Lu, and X. Wang. Unveiling encoder-free vision-language models. *arXiv preprint arXiv:2406.11832*, 2024.

A. Drouin, M. Gasse, M. Caccia, I. H. Laradji, M. D. Verme, T. Marty, L. Boisvert, M. Thakkar, Q. Cappart, D. Vazquez, N. Chapados, and A. Lacoste. Workarena: How capable are web agents at solving common knowledge work tasks?, 2024. URL `https://arxiv.org/abs/2403.07718`.

H. Duan, J. Yang, Y. Qiao, X. Fang, L. Chen, Y. Liu, X. Dong, Y. Zang, P. Zhang, J. Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201, 2024.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, and et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Roziere, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Wyatt, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Guzmán, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Thattai, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. Kloumann, I. Misra, I. Evtimov, J. Zhang, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Prasad, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, K. El-Arini, K. Iyer, K. Malik, K. Chiu, K. Bhalla, K. Lakhotia, L. Rantala-Yeary, L. van der Maaten, L. Chen, L. Tan, L. Jenkins, L. Martin, L. Madaan, L. Malo, L. Blecher, L. Landzaat, L. de Oliveira, M. Muzzi, M. Pasupuleti, M. Singh, M. Paluri, M. Kardas, M. Tsimpoukelli, M. Oldham, M. Rita, M. Pavlova, M. Kambadur, M. Lewis, M. Si, M. K. Singh, M. Hassan, N. Goyal, N. Torabi, N. Bashlykov, N. Bogoychev, N. Chatterji, N. Zhang, O. Duchenne, O. Çelebi, P. Alrassy, P. Zhang, P. Li, P. Vasic, P. Weng, P. Bhargava, P. Dubal, P. Krishnan, P. S. Koura, P. Xu, Q. He, Q. Dong, R. Srinivasan, R. Ganapathy, R. Calderer, R. S. Cabral, R. Stojnic, R. Raileanu, R. Maheswari, R. Girdhar, R. Patel, R. Sauvestre, R. Polidoro, R. Sumbaly, R. Taylor, R. Silva, R. Hou, R. Wang, S. Hosseini, S. Chennabasappa, S. Singh, S. Bell, S. S. Kim, S. Edunov, S. Nie, S. Narang, S. Raparthy, S. Shen, S. Wan, S. Bhosale, S. Zhang, S. Vandenhende, S. Batra, S. Whitman, S. Sootla, S. Collot, S. Gururangan, S. Borodinsky, T. Herman, T. Fowler, T. Sheasha, T. Georgiou, T. Scialom, T. Speckbacher, T. Mihaylov, T. Xiao, U. Karn, V. Goswami, V. Gupta, V. Ramanathan, V. Kerkez, V. Gonguet, V. Do, V. Vogeti, V. Albiero, V. Petrovic, W. Chu, W. Xiong, W. Fu, W. Meers, X. Martinet, X. Wang, X. Wang, X. E. Tan, X. Xia, X. Xie, X. Jia, X. Wang, Y. Goldschlag, Y. Gaur, Y. Babaei, Y. Wen, Y. Song, Y. Zhang, Y. Li, Y. Mao, Z. D. Coudert, Z. Yan, Z. Chen, Z. Papakipos, A. Singh, A. Srivastava, A. Jain, A. Kelsey, A. Shajnfeld, A. Gangidi, A. Victoria, A. Goldstand, A. Menon, A. Sharma, A. Boesenberg, A. Baevski, A. Feinstein, A. Kallet, A. Sangani, A. Teo, A. Yunus, A. Lupu, A. Alvarado, A. Caples, A. Gu, A. Ho, A. Poulton, A. Ryan, A. Ramchandani, A. Dong, A. Franco, A. Goyal, A. Saraf, A. Chowdhury, A. Gabriel, A. Bharambe, A. Eisenman, A. Yazdan, B. James, B. Maurer, B. Leonhardi, B. Huang, B. Loyd, B. D. Paola, B. Paranjape, B. Liu, B. Wu, B. Ni, B. Hancock, B. Wasti, B. Spence, B. Stojkovic, B. Gamido, B. Montalvo, C. Parker, C. Burton, C. Mejia, C. Liu, C. Wang, C. Kim, C. Zhou, C. Hu, C.-H. Chu, C. Cai, C. Tindal, C. Feichtenhofer, C. Gao, D. Civin, D. Beaty, D. Kreymer, D. Li, D. Adkins, D. Xu, D. Testuggine, D. David, D. Parikh, D. Liskovich, D. Foss, D. Wang, D. Le, D. Holland, E. Dowling, E. Jamil, E. Montgomery, E. Presani, E. Hahn, E. Wood, E.-T. Le, E. Brinkman, E. Arcaute, E. Dunbar, E. Smothers, F. Sun, F. Kreuk, F. Tian, F. Kokkinos, F. Ozgenel, F. Caggioni, F. Kanayet, F. Seide, G. M. Florez, G. Schwarz, G. Badeer, G. Swee, G. Halpern, G. Herman, G. Sizov, Guangyi, Zhang, G. Lakshminarayanan, H. Inan, H. Shojanazeri, H. Zou, H. Wang, H. Zha, H. Habeeb, H. Rudolph, H. Suk, H. Aspegren, H. Goldman, H. Zhan, I. Damlaj, I. Molybog, I. Tufanov, I. Leontiadis, I.-E. Veliche, I. Gat, J. Weissman, J. Geboski, J. Kohli, J. Lam, J. Asher, J.-B. Gaya, J. Marcus, J. Tang, J. Chan, J. Zhen, J. Reizenstein, J. Teboul, J. Zhong, J. Jin, J. Yang, J. Cummings, J. Carvill, J. Shepard, J. McPhie, J. Torres, J. Ginsburg, J. Wang, K. Wu, K. H. U, K. Saxena, K. Khandelwal, K. Zand, K. Matosich, K. Veeraraghavan, K. Michelena, K. Li, K. Jagadeesh, K. Huang, K. Chawla, K. Huang, L. Chen, L. Garg, L. A, L. Silva, L. Bell, L. Zhang, L. Guo, L. Yu, L. Moshkovich, L. Wehrstedt, M. Khabsa, M. Avalani, M. Bhatt, M. Mankus, M. Hasson, M. Lennie, M. Reso, M. Groshev, M. Naumov, M. Lathi, M. Keneally, M. Liu, M. L. Seltzer, M. Valko, M. Restrepo, M. Patel, M. Vyatskov, M. Samvelyan, M. Clark, M. Macey, M. Wang, M. J. Hermoso, M. Metanat, M. Rastegari, M. Bansal, N. Santhanam, N. Parks, N. White, N. Bawa, N. Singhal, N. Egebo, N. Usunier, N. Mehta, N. P. Laptev, N. Dong, N. Cheng, O. Chernoguz, O. Hart, O. Salpekar, O. Kalinli, P. Kent, P. Parekh, P. Saab, P. Balaji, P. Rittner, P. Bontrager, P. Roux, P. Dollar, P. Zvyagina, P. Ratanchandani, P. Yuvraj, Q. Liang, R. Alao, R. Rodriguez, R. Ayub, R. Murthy, R. Nayani, R. Mitra, R. Parthasarathy, R. Li, R. Hogan, R. Battey, R. Wang, R. Howes, R. Rinott, S. Mehta, S. Siby, S. J. Bondu, S. Datta, S. Chugh, S. Hunt, S. Dhillon, S. Sidorov, S. Pan, S. Mahajan, S. Verma, S. Yamamoto, S. Ramaswamy, S. Lindsay, S. Lindsay, S. Feng, S. Lin, S. C. Zha, S. Patil, S. Shankar, S. Zhang, S. Zhang, S. Wang, S. Agarwal, S. Sajuyigbe, S. Chintala, S. Max, S. Chen, S. Kehoe, S. Satterfield, S. Govindaprasad,

S. Gupta, S. Deng, S. Cho, S. Virk, S. Subramanian, S. Choudhury, S. Goldman, T. Remez, T. Glaser, T. Best, T. Koehler, T. Robinson, T. Li, T. Zhang, T. Matthews, T. Chou, T. Shaked, V. Vontimitta, V. Ajayi, V. Montanez, V. Mohan, V. S. Kumar, V. Mangla, V. Ionescu, V. Poenaru, V. T. Mihailescu, V. Ivanov, W. Li, W. Wang, W. Jiang, W. Bouaziz, W. Constable, X. Tang, X. Wu, X. Wang, X. Wu, X. Gao, Y. Kleinman, Y. Chen, Y. Hu, Y. Jia, Y. Qi, Y. Li, Y. Zhang, Y. Zhang, Y. Adi, Y. Nam, Yu, Wang, Y. Zhao, Y. Hao, Y. Qian, Y. Li, Y. He, Z. Rait, Z. DeVito, Z. Rosnbrick, Z. Wen, Z. Yang, Z. Zhao, and Z. Ma. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. URL `https://arxiv.org/abs/1512.03385`.

A. Hu, H. Xu, J. Ye, M. Yan, L. Zhang, B. Zhang, C. Li, J. Zhang, Q. Jin, F. Huang, and J. Zhou. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding, 2024. URL `https://arxiv.org/abs/2403.12895`.

D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, 2019. URL `https://arxiv.org/abs/1902.09506`.

G. Jaume, H. K. Ekenel, and J.-P. Thiran. Funsd: A dataset for form understanding in noisy scanned documents, 2019. URL `https://arxiv.org/abs/1905.13538`.

G. Kim, T. Hong, M. Yim, J. Nam, J. Park, J. Yim, W. Hwang, S. Yun, D. Han, and S. Park. Ocr-free document understanding transformer, 2022. URL `https://arxiv.org/abs/2111.15664`.

Y. Kim, M. Yim, and K. Y. Song. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv preprint arXiv:2404.19205*, 2024.

H. Laurençon, L. Tronchon, M. Cord, and V. Sanh. What matters when building vision-language models?, 2024. URL `https://arxiv.org/abs/2405.02246`.

K. Lee, M. Joshi, I. Turc, H. Hu, F. Liu, J. Eisenschlos, U. Khandelwal, P. Shaw, M.-W. Chang, and K. Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding, 2023. URL `https://arxiv.org/abs/2210.03347`.

B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension, 2023a. URL `https://arxiv.org/abs/2307.16125`.

B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, and C. Li. Llava-onevision: Easy visual task transfer, 2024. URL `https://arxiv.org/abs/2408.03326`.

J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023b. URL `https://arxiv.org/abs/2301.12597`.

Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, and J.-R. Wen. Evaluating object hallucination in large vision-language models, 2023c. URL `https://arxiv.org/abs/2305.10355`.

H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning, 2023a.

H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning, 2023b.

H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

S. Lu, Y. Li, Q.-G. Chen, Z. Xu, W. Luo, K. Zhang, and H.-J. Ye. Ovis: Structural embedding alignment for multimodal large language model, 2024. URL `https://arxiv.org/abs/2405.20797`.

A. Masry, D. X. Long, J. Q. Tan, S. Joty, and E. Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

M. Mathew, V. Bagal, R. P. Tito, D. Karatzas, E. Valveny, and C. V. Jawahar. Infographicvqa, 2021a. URL `https://arxiv.org/abs/2104.12756`.

M. Mathew, D. Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021b. URL `https://arxiv.org/abs/2007.00398`.

OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, et al. Gpt-4 technical report. *arXiv preprint arXiv: 2303.08774*, 2023.

S. Park, S. Shin, B. Lee, J. Lee, J. Surh, M. Seo, and H. Lee. Cord: A consolidated receipt dataset for post-ocr parsing. *Document Intelligence Workshop at Neural Information Processing Systems*, 2019.

P. Pasupat and P. Liang. Compositional semantic parsing on semi-structured tables. In *Annual Meeting of the Association for Computational Linguistics*, 2015.

Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021. URL `https://arxiv.org/abs/2103.00020`.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL `https://arxiv.org/abs/1910.10683`.

J. Rodriguez, X. Jian, S. S. Panigrahi, T. Zhang, A. Feizi, A. Puri, A. Kalkunte, F. Savard, A. Masry, S. Nayak, R. Awal, M. Massoud, A. Abaskohi, Z. Li, S. Wang, P.-A. Noël, M. L. Richter, S. Vadacchino, S. Agarwal, S. Biswas, S. Shanian, Y. Zhang, N. Bolger, K. MacDonald, S. Fauvel, S. Tejaswi, S. Sunkara, J. Monteiro, K. D. Dvijotham, T. Scholak, N. Chapados, S. Kharagani, S. Hughes, M. Özsu, S. Reddy, M. Pedersoli, Y. Bengio, C. Pal, I. Laradji, S. Gella, P. Taslakian, D. Vazquez, and S. Rajeswar. Bigdocs: An open and permissively-licensed dataset for training multimodal models on document and code tasks, 2024a. URL `https://arxiv.org/abs/2412.04626`.

J. A. Rodriguez, D. Vazquez, I. Laradji, M. Pedersoli, and P. Rodriguez. Ocr-vqgan: Taming text-within-image generation, 2022. URL `https://arxiv.org/abs/2210.11248`.

J. A. Rodriguez, A. Puri, S. Agarwal, I. H. Laradji, P. Rodriguez, S. Rajeswar, D. Vazquez, C. Pal, and M. Pedersoli. Starvector: Generating scalable vector graphics code from images and text, 2024b. URL `https://arxiv.org/abs/2312.11556`.

A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards vqa models that can read. In *IEEE Conference Computer Vision Pattern Recognition*, 2019.

T. Stanisławek, F. Graliński, A. Wróblewska, D. Lipiński, A. Kaliska, P. Rosalska, B. Topolski, and P. Biecek. Kleister: key information extraction datasets involving long documents with complex layouts. In *International Conference on Document Analysis and Recognition*, 2021.

S. Svetlichnaya. Deepform: Understand structured documents at scale, 2020.

G. Team. Gemini: A family of highly capable multimodal models, 2024. URL `https://arxiv.org/abs/2312.11805`.

C. Vogus and E. Llansóe. Making transparency meaningful: A framework for policymakers. *Center for Democracy and Technology*, 2021.

D. Wang, N. Raman, M. Sibue, Z. Ma, P. Babkin, S. Kaur, Y. Pei, A. Nourbakhsh, and X. Liu. Docllm: A layout-aware generative language model for multimodal document understanding, 2023a. URL `https://arxiv.org/abs/2401.00908`.

P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. URL `https://arxiv.org/abs/2409.12191`.

W. Wang, Q. Lv, W. Yu, W. Hong, J. Qi, Y. Wang, J. Ji, Z. Yang, L. Zhao, X. Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023b.

C. Wu, X. Chen, Z. Wu, Y. Ma, X. Liu, Z. Pan, W. Liu, Z. Xie, X. Yu, C. Ruan, and P. Luo. Janus: Decoupling visual encoding for unified multimodal understanding and generation, 2024a. URL `https://arxiv.org/abs/2410.13848`.

Z. Wu, X. Chen, Z. Pan, X. Liu, W. Liu, D. Dai, H. Gao, Y. Ma, C. Wu, B. Wang, Z. Xie, Y. Wu, K. Hu, J. Wang, Y. Sun, Y. Li, Y. Piao, K. Guan, A. Liu, X. Xie, Y. You, K. Dong, X. Yu, H. Zhang, L. Zhao, Y. Wang, and C. Ruan. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding, 2024b. URL `https://arxiv.org/abs/2412.10302`.

R. Xu, Y. Yao, Z. Guo, J. Cui, Z. Ni, C. Ge, T.-S. Chua, Z. Liu, M. Sun, and G. Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. *European Conference on Computer Vision*, 2024. doi: 10.48550/arXiv.2403.11703.

W. Yu, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, X. Wang, and L. Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities, 2024. URL `https://arxiv.org/abs/2308.02490`.

X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, C. Wei, B. Yu, R. Yuan, R. Sun, M. Yin, B. Zheng, Z. Yang, Y. Liu, W. Huang, H. Sun, Y. Su, and W. Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. URL `https://arxiv.org/abs/2311.16502`.

X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training, 2023. URL `https://arxiv.org/abs/2303.15343`.

T. Zhang, S. Wang, L. Li, G. Zhang, P. Taslakian, S. Rajeswar, J. Fu, B. Liu, and Y. Bengio. Vcr: Visual caption restoration. *arXiv preprint arXiv: 2406.06462*, 2024.

Y. Zhao, J. Huang, J. Hu, X. Wang, Y. Mao, D. Zhang, Z. Jiang, Z. Wu, B. Ai, A. Wang, W. Zhou, and Y. Chen. Swift:a scalable lightweight infrastructure for fine-tuning, 2024. URL `https://arxiv.org/abs/2408.05517`.

J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao, Z. Gao, E. Cui, X. Wang, Y. Cao, Y. Liu, X. Wei, H. Zhang, H. Wang, W. Xu, H. Li, J. Wang, N. Deng, S. Li, Y. He, T. Jiang, J. Luo, Y. Wang, C. He, B. Shi, X. Zhang, W. Shao, J. He, Y. Xiong, W. Qu, P. Sun, P. Jiao, H. Lv, L. Wu, K. Zhang, H. Deng, J. Ge, K. Chen, L. Wang, M. Dou, L. Lu, X. Zhu, T. Lu, D. Lin, Y. Qiao, J. Dai, and W. Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL `https://arxiv.org/abs/2504.10479`.

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: All the claims and contributions presented in the abstract and introduction are discussed in Section 3 (Methodology) and supported by results in Section 5.

   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: Limitations of our approach are discussed in the results section of the paper (Section 5).

   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: [NA]

   Justification: The paper does not discuss theoretical results. The main contributions and claims are supported by empirical results.

   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: All the details necessary to reproduce our results are provided in the Methodology (Section 3), Experimental Setup (Section 4), and Appendix A.1. Additionally, our code will be made available upon acceptance.

   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**

   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: We will provide full access to our code upon the acceptance of the paper.

   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: All the details necessary to reproduce our results are provided in the Methodology (Section 3), Experimental Setup (Section 4), and Appendix A.1. Additionally, our code will be made available upon acceptance.

   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: Error bars are not reported due to the expensive computational requirements to produce them.

   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).
   - 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: Information regrading the computing resources are provided in the Experimental Setup (Section 4).

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: The work conform with the NeurIPS Code of Ethics.

   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: [NA]

    Justification: The work in this paper has no societal impact.

    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.
    - 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: [NA]

    Justification: Our work does not pose any high-level risks.

    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 cite the corresponding papers of the models and datasets that we use in our experiments. In addition, our work adhere to their terms of use and licenses.

    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.
    - 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: Our codes and assets will be released upon paper acceptance.

    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 work in this 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: This work does not involve crowdsourcing or research involving 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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were not used in a capacity that could be considered important, original, or as a non-standard component of the work.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Appendix

A.1 Experimental Setup

We provide detailed hyperparameters of our experiments in Table 6.

Table 6: Detailed hyperparameters for each training stage across different LLM backbones.

LLM Backbone	Llama 3.2-1B			Llama 3.2-3B			Llama 3.1-8B		
	Stage-1	Stage-2	Stage-3	Stage-1	Stage-2	Stage-3	Stage-1	Stage-2	Stage-3
Trainable Parameters	Full Model	Full Model	LLM & Connector	Full Model	Full Model	LLM & Connector	Full Model	Full Model	LLM & Connector
Batch Size	512	512	512	512	256	256	512	256	256
Text Max Length	1024	2048	2048	1024	2048	2048	1024	2048	2048
Epochs	1	1	5	1	1	5	1	1	5
Learning Rate	$1 \;\; 10^{5}$	$5 \;\; 10^{5}$	$5 \;\; 10^{5}$	$1 \;\; 10^{5}$	$5 \;\; 10^{5}$	$5 \;\; 10^{5}$	$1 \;\; 10^{5}$	$1 \;\; 10^{5}$	$1 \;\; 10^{5}$

A.2 Runtime Comparison Between Connectors

One caveat in the ALIGN connector is that it includes an additional LM head layer, which slightly increases the total number of parameters. However, this addition has a negligible impact on runtime efficiency due to its simple structure. It only introduces a few matrix multiplication operations (as shown in Equations 1 and 2) instead of stacking many complex layers that require sequential processing, as in deep fusion methods.

To empirically validate this claim, we benchmarked the runtime and memory usage of models equipped with different connector types (`MLP`, `Align`, `Ovis`, and `Perceiver`), following the same experimental setup as in Table 2. As shown in Table 7, the results demonstrate that although the ALIGN connector delivers notably superior performance (see Table 2), the variations in inference speed and GPU memory usage among the connectors remain minimal.

Table 7: Runtime and memory comparison between different connector designs. The results show that ALIGN introduces negligible computational overhead compared to other connectors.

Model	Samples	Avg Time (s)	Tokens/sec	GPU Memory (GB)
Llama-3.2-3B-MLP	2500	0.161	118.3	10.9
Llama-3.2-3B-Perceiver	2500	0.140	135.1	10.9
Llama-3.2-3B-Ovis	2500	0.155	122.5	10.8
Llama-3.2-3B-ALIGN	2500	0.165	115.4	10.9

Overall, the empirical evidence confirms that the ALIGN connector achieves an effective balance between computational efficiency and performance. It introduces only a negligible increase in runtime and memory usage while providing substantial gains in overall accuracy.

A.3 Pixel-Level Tasks Analysis

To rigorously evaluate the ability of vision-language models to integrate fine-grained visual and textual pixel-level cues, we test our model on the VCR benchmark [Zhang et al., 2024], which requires the model to recover partially occluded texts with pixel-level hints from the revealed parts of the text. This task challenges VLM's alignment of text and image in extreme situations. Current state-of-the-art models like GPT-4V OpenAI et al. [2023], Claude 3.5 Sonnet Anthropic [2024], and Llama-3.2 Dubey et al. [2024] significantly underperform humans on *hard* VCR task due to their inability to process subtle pixel-level cues in occluded text regions. These models frequently discard critical visual tokens during image tokenization on semantic priors, overlooking the interplay between partial character strokes and contextual visual scenes. To evaluate performance on VCR, we modify our Stage 3 SFT dataset composition by replacing the exclusive use of DocDownstream with a 5:1 blended ratio of DocDownstream and VCR training data. This adjustment enables direct evaluation of our architecture ALIGN's ability to leverage pixel-level character cues.

From the experimental outcomes, it is evident that ALIGNVLM consistently outperforms the MLP Connector Model across both easy and hard settings of the pixel-level VCR task (see Figure 5), with improvements ranging from 10.18% on the hard setting to 14.41% on the easy setting.

We provide a case study on VCR in Figure 6, featuring four representative examples. In Figure 6a, it is evident that the MLP connector model fails to capture semantic consistency as effectively as ALIGNVLM. The phrase "The commune first *census in written history in*" (where the words in italics are generated by the model while the rest are in the image) is not as semantically coherent as the phrase generated by ALIGN "The commune first *appears in written history in*".

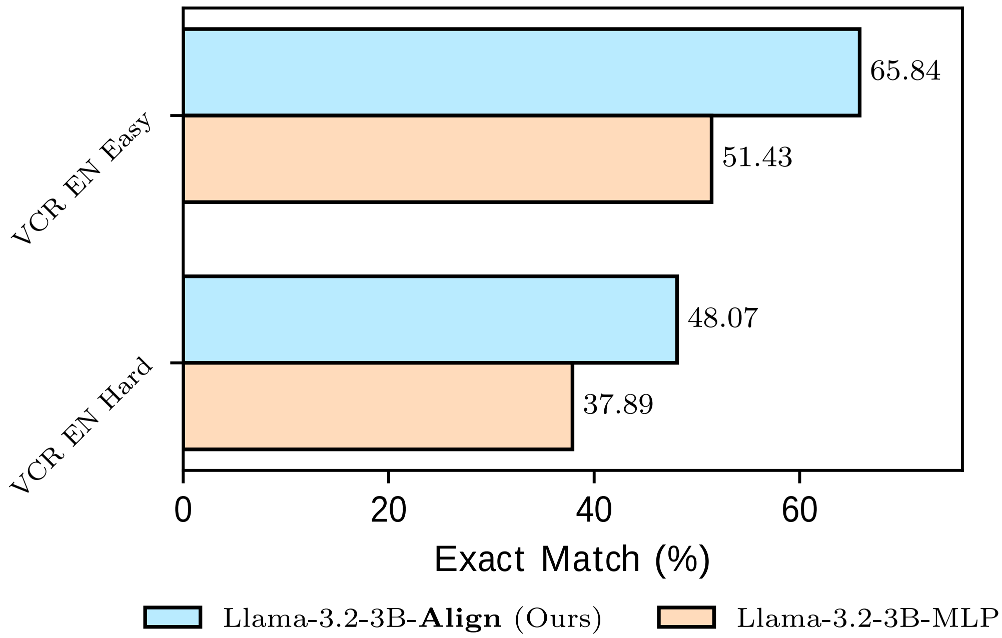


Figure 5: Comparison of Llama-3.2-3b-ALIGN and Llama-3.2-3B-MLP on the Easy and Hard VCR tasks.

Beyond the issue of semantic fluency, in Figure 6b we also observe that ALIGNVLM successfully identifies the uncovered portion of the letter “g” in “accounting” and uses it as a pixel-level hint to infer the correct word. In contrast, the MLP model fails to effectively attend to this crucial detail.

Figures 6c and 6d show examples where ALIGNVLM fails on the VCR task. These carefully picked instances show that our method mistakes names of landmarks with common words when the two are very similar. As seen in the examples, ALIGNVLM mistakes “Llanengan" for “Llanongan" and “Gorden" for “Garden”. In both instances, the pairs differ by one character, indicating perhaps that ALIGNVLM tends to align vision representations to more common tokens in the vocabulary. One approach that would potentially mitigate such errors would be to train ALIGNVLM with more contextually-relevant data.

A.4 Case Studies

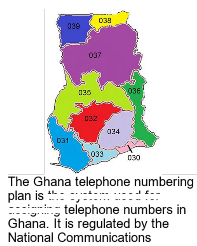
In this section, we provide case studies for the experiments in Section 5.1. Specifically, we provide examples of our Llama-3.2-3B-ALIGN, and its counterpart model with alternative connectors Llama-3.2-3B-MLP and Llama-3.2-3B-Ovis on three different datasets: KLC [Stanisławek et al., 2021], DocVQA [Mathew et al., 2021b], and TextVQA [Singh et al., 2019]. The examples are shown in Figure 7, 8, and 9.

GT: *(appears in written history in)*

MLP: *(census in written history in)* 7

ALIGN *(appears in written history in)* 3

(a) Positive Example 1

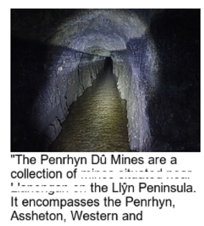


GT: *(the system used for assigning)*

MLP: *(the system used for accounting)* 7

ALIGN *(the system used for assigning)* 3

(b) Positive Example 2



GT: *(mines situated near Llanengan on)*

MLP: *(mines situated near Llanengan on)* 3

ALIGN *(mines situated near Llanongan on)* 7

(c) Negative Example 1

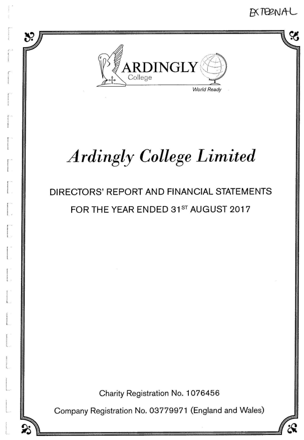
GT: *(Gorden County is home to)*

MLP: *(Gorden County is home to)* 3

ALIGN *(Garden County is home to)* 7

(d) Negative Example 2

Figure 6: **Case Study for Pixel-Level Tasks.** We provide examples of our proposed ALIGN connector compared with a the Multi-Layer Perceptron (MLP) connector. The ALIGN connector tends to better map visual elements to common words. GT is the ground truth.


EXTERNAL

ARDINGLY College

World Ready

Ardingly College Limited

DIRECTORS' REPORT AND FINANCIAL STATEMENTS

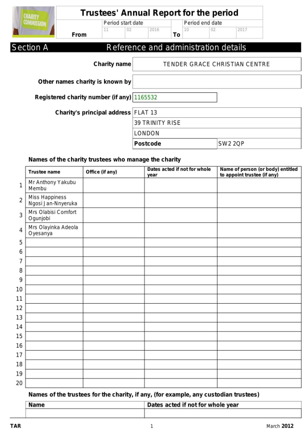
FOR THE YEAR ENDED 31ST AUGUST 2017

Charity Registration No. 1076456

Company Registration No. 03779971 (England and Wales)

Question: What is the value for the charity name?
GT: *(Ardingly College Ltd.)*
MLP: *(Ardington College Ltd.)* 7
Ovis: *(Ardington College Ltd.)* 7
ALIGN: *(Ardingly College Ltd.)* 3

(a) Positive Example #1


Trustees' Annual Report for the period

Section A Reference and administration details

Charity name	TENDER GRACE CHRISTIAN CENTRE
Other names charity is known by	
Registered charity number (if any)	1165532
Charity's principal address	FLAT 13 39 TRINITY RISE LONDON Postcode SW2 2QP

Names of the charity trustees who manage the charity

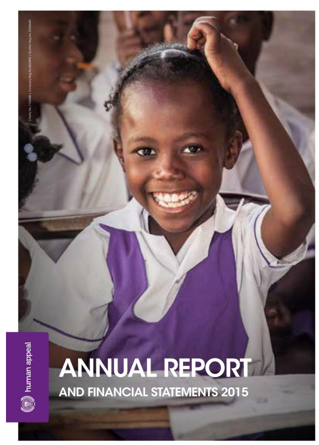
	Trustee name	Office (if any)	Dates acted if not for whole year	Name of person (or body) entitled to appoint trustee (if any)
1	Mr Anthony Yakubu Membu			
2	Miss Happiness Ngozi Jan-Nnyeruka			
3	Mrs Olabisi Comfort Ogunjobi			
4	Mrs Olayinka Adeola Oyesanya			

Names of the trustees for the charity, if any, (for example, any custodian trustees)

TAR 1 March 2012

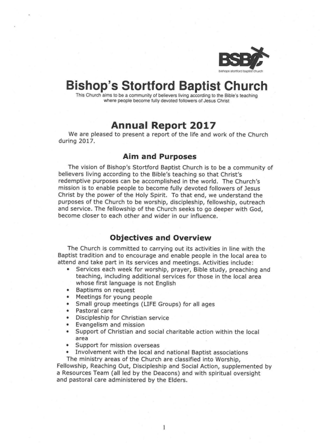
Question: What is the value for the address postcode?
GT: *(SW2 2QP)*
MLP: *(SW22 0PQ)* 7
Ovis: *(SW2 2OP)* 7
ALIGN: *(SW2 2QP)* 3

(b) Positive Example #2



Question: What is the value for the charity name?
GT: *(Human Appeal)*
MLP: *(Humanitarian Agenda)* 7
Ovis: *(Human Appeal)* 3
ALIGN: *(Human Rightsappeal)* 7

(c) Negative Example #1


BSBC

Bishop's Stortford Baptist Church

Annual Report 2017

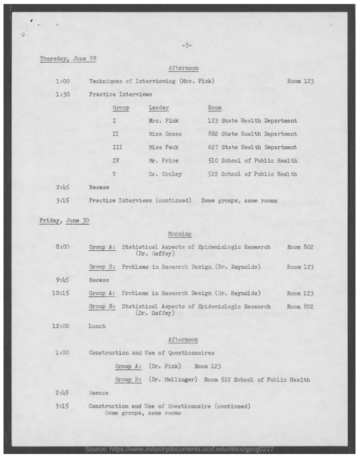
Aim and Purposes

Objectives and Overview

Question: What is the value for the post town address?
GT: *(Bishop's Stortford)*
MLP: *(Stortford)* 7
Ovis: *(Bishop's Stortford)* 3
ALIGN: *(Stortford)* 7

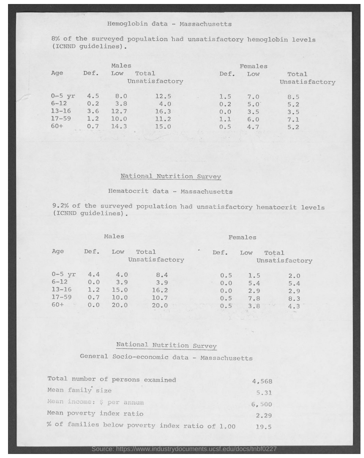
(d) Negative Example #2

Figure 7: **Case Study for Connector Comparison on the KLC dataset [Stanisławek et al., 2021].** We show four qualitative examples (including two correct and two incorrect examples) comparing Llama-3.2-3B-ALIGN to the same architecture with different connectors, Llama-3.2-3B-MLP and Llama-3.2-3B-Ovis. "GT" denotes the ground truth.



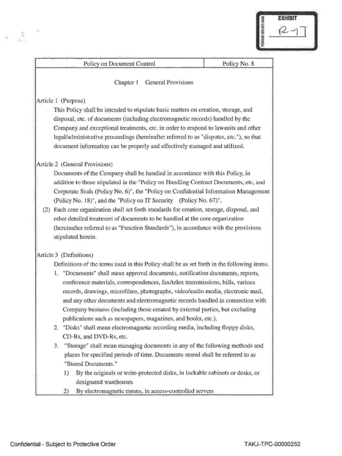
Question: What does the afternoon session begin on June 29?
GT: *(1:00)*
MLP: *(2:45)* 7
Ovis: *(3:30)* 7
ALIGN: *(1:00)* 3

(a) Positive Example #1



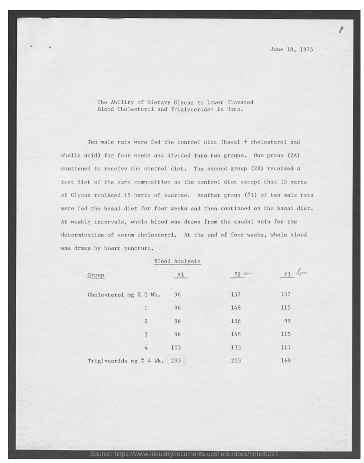
Question: What levels does the second table indicate?
GT: *(hematocrit data - Massachusetts)*
MLP: *(SATISFACTORY)* 7
Ovis: *(Females)* 7
ALIGN: *(hematocrit data - Massachusetts)* 3

(b) Positive Example #2



Question: What type of policy is described in this document?
GT: *(Policy on Document Control)*
MLP: *(Policy on Document Control)* 3
Ovis: *(General Provisions)* 7
ALIGN: *(Document Control)* 7

(c) Negative Example #1



Question: What was the diet fed to the #1 group?
GT: *(basal diet)*
MLP: *(basel diet)* 3
Ovis: *(Whole blood)* 7
ALIGN: *(control diet)* 7

(d) Negative Example #2

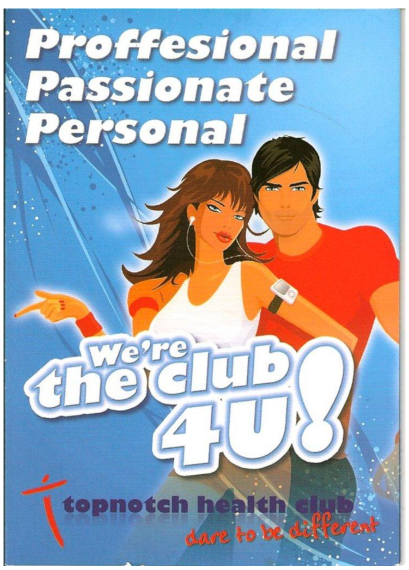
Figure 8: **Case Study for Connector Comparison on the DocVQA dataset [Mathew et al., 2021b].** We show four qualitative examples (including two correct and two incorrect examples) comparing Llama-3.2-3B-ALIGN to the same architecture with different connectors, Llama-3.2-3B-MLP and Llama-3.2-3B-Ovis. "GT" denotes the ground truth.

Question: What greeting is written on the letter?
GT: *(good bye)*
MLP: *(good)* 7
Ovis: *(good buy)* 7
ALIGN: *(good bye)* 3

(a) Positive Example #1

Question: What indoor temperature is shown?
GT: *(68.4)*
MLP: *(68 F)* 7
Ovis: *(40.0)* 7
ALIGN: *(68.4)* 3

(b) Positive Example #2



Question: What type of club is advertised?
GT: *(health club)*
MLP: *(topnote health club)* 7
Ovis: *(health club)* 3
ALIGN: *(professional passionate personal)* 7

(c) Negative Example #1

Question: What credit card is this?
GT: *(hadiah plus)*
MLP: *(hadiah plus)* 3
Ovis: *(american big loyalty program)* 7
ALIGN: *(hadia plus)* 7

(d) Negative Example #2

Figure 9: **Case Study for Connector Comparison on the TextVQA dataset [Singh et al., 2019].** We show four qualitative examples (including two correct and two incorrect examples) comparing Llama-3.2-3B-ALIGN to the same architecture with different connectors, Llama-3.2-3B-MLP and Llama-3.2-3B-Ovis. "GT" denotes the ground truth.