# Dense Connector for MLLMs

**Huanjin Yao**[1,3*], **Wenhao Wu**[2*✉], **Taojiannan Yang**[4], **Yuxin Song**[3], **Mengxi Zhang**[3]
**Haocheng Feng**[3], **Yifan Sun**[3], **Zhiheng Li**[1✉], **Wanli Ouyang**[5], **Jingdong Wang**[3]
[1]Shenzhen International Graduate School, Tsinghua University [2]The University of Sydney
[3]Baidu Inc. [4]Amazon [5] The Chinese University of Hong Kong
[*] Equal Contribution ✉ Corresponding Author

## Abstract

*Do we fully leverage the potential of visual encoder in Multimodal Large Language Models (MLLMs)?* The recent outstanding performance of MLLMs in multimodal understanding has garnered broad attention from both academia and industry. In the current MLLM rat race, the focus seems to be predominantly on the linguistic side. We witness the rise of larger and higher-quality instruction datasets, as well as the involvement of larger-sized LLMs. Yet, scant attention has been directed towards the visual signals utilized by MLLMs, often assumed to be the final high-level features extracted by a frozen visual encoder. In this paper, we introduce the ***Dense Connector*** - a simple, effective, and plug-and-play vision-language connector that significantly enhances existing MLLMs by leveraging multi-layer visual features, with minimal additional computational overhead. Building on this, we also propose the Efficient Dense Connector, which achieves performance comparable to LLaVA-v1.5 with only 25% of the visual tokens. Furthermore, our model, trained solely on images, showcases remarkable zero-shot capabilities in video understanding as well. Experimental results across various vision encoders, image resolutions, training dataset scales, varying sizes of LLMs (2.7B→70B), and diverse architectures of MLLMs (*e.g.*, LLaVA-v1.5, LLaVA-NeXT and Mini-Gemini) validate the versatility and scalability of our approach, achieving state-of-the-art performance across 19 image and video benchmarks. We hope that this work will provide valuable experience and serve as a basic module for future MLLM development. Code is available at `https://github.com/HJYao00/DenseConnector`.

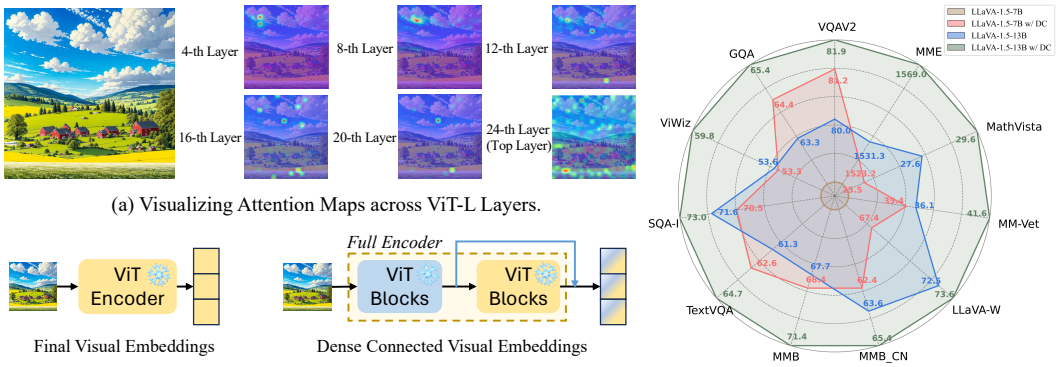
## 1 Introduction

In recent years, Large Language Models (LLMs) led by ChatGPT [1] have made remarkable advancements in text comprehension and generation. Furthermore, cutting-edge Multimodal Large Language Models (MLLMs) [2, 3] have rapidly expanded the capabilities of LLMs to include visual understanding, evolving into models capable of integrating both vision and text modalities. This has elevated MLLMs to become a new focal point for research and discussion [4, 5, 6, 7].

In broad terms, the architecture of existing MLLMs can be delineated into three components: the pre-trained vision encoder (*e.g.*, CLIP's ViT-L [8] or EVA-CLIP's ViT-G [9]), the pre-trained LLM (*e.g.*, OPT [10], Llama [11], Vicuna [12], *etc.*), and the connector (*e.g.*, Q-former [13, 14] or linear projection [15, 16]) trained from scratch to bridge the vision and language models. An intriguing trend in current MLLM research is that the focus of model learning and performance improvement seems to primarily center around the language aspect (*e.g.*, utilizing larger-scale and higher-quality visual instruction data [17, 16, 18], larger-sized LLMs [19, 20]), with less exploration into the visual signals fed into the connector. Typically, the visual encoder is frozen to extract high-level visual features, which are then fed into the connector. This leads us to rethink: *Have we fully utilized the existing pre-trained visual encoder?*

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-1043




(a) Visualizing Attention Maps across ViT-L Layers.

(b) Visual Embeddings: Final *vs.* Dense Connectivity in MLLMs

(c) Plug-and-Play Dense Connector

Figure 1: Exploring Multi-layer Visual Features Empowering existing MLLMs.

In addition to the common practice of feeding the connector with final high-level visual features from visual encoder, an intuitive yet overlooked idea is to integrate visual features from various layers to complement the high-level features. In Fig. 1 (a), we illustrate attention maps from different layers of a 24-layer CLIP [8] pre-trained ViT-L [21], showing that different layers of the same visual encoder emphasize different regions of interest. Moreover, looking back at the history of computer vision, classic models (*e.g.*, Densenet [22], FPN [23]) utilize multi-layer features to enhance visual representations. In MLLMs, the vision encoder is typically frozen to mitigate significant computational costs. In this context, our idea leverages the "free lunch" of utilizing offline features from different layers as an implicit enhancement of visual information without the need for additional computational overhead. Furthermore, this way also complements techniques that directly increase visual signals, *e.g.*, increasing image resolution [18, 24, 25, 26, 27, 28] or introducing additional visual encoders [29, 30, 18]. This idea is both simple and efficient, while also being sufficiently generic, logically allowing for seamless integration with any existing MLLMs.

In light of this, we propose the ***Dense Connector (DC)***, serving as a plug-and-play vision-language connector that involves offline features from various layers of the frozen visual encoder to provide the LLM with more visual cues. We explore three intuitive instantiations for the Dense Connector: 1) *Sparse Token Integration (STI)*: We explicitly consider increasing the number of visual tokens by aggregating visual tokens from different specified layers and the final visual token. These tokens are then fed into a learnable projector for mapping into the text space. 2) *Sparse Channel Integration (SCI)*: To avoid increasing the number of tokens, we concatenate visual tokens from different specified layers in the feature dimension. They are then passed to the projector, which not only maps visual tokens into the text space but also serves to reduce the feature dimensionality. 3) *Dense Channel Integration (DCI)*: In addition to incorporating features from specified layers, we further attempt to utilize visual features from all layers. All of these instantiations yield significant improvements while utilizing just one simple learnable projector (comprising two linear layers) without introducing any extra parameters. Moreover, we conduct extensive empirical studies to demonstrate its scalability and compatibility. We summarize our contributions as follows:

- We propose a simple, effective, and plug-and-play *Dense Connector* that enhances the visual representation of existing MLLMs with minimal additional computational overhead. We hope it can serve as a basic module to continuously benefit future MLLMs.
- We demonstrate the versatility and scalability of our approach across various visual encoders, image resolutions (336px→768px), training dataset scales, varying sizes of LLMs (2B→70B), and diverse MLLMs architectures (*e.g.*, LLaVA-v1.5 [16], LLaVA-NeXT [25], Mini-Gemini [18]).
- Our method exhibits exceptional performance across 11 image benchmarks and achieves state-of-the-art results on 8 video benchmarks without the need for specific video tuning.

https://doi.org/10.52202/079017-1043

## 2 Related Work

### 2.1 Large Pre-trained Vision Models

The advent of pre-trained Vision Transformers (ViT) [21] has significantly propelled the advancement of computer vision. Furthermore, pre-training ViT models on web-scale image-text pairs, *e.g.*, CLIP [8] and its subsequent iterations [9, 31, 32, 33], where vision and text encoders are simultaneously trained to bridge the gap between visual and textual modalities, has introduced zero-shot visual perception capabilities. Since then, CLIP-like models have served as effective initializations and have been incorporated into various vision-language cross-modal models (*e.g.*, video-text alignment [34, 35, 36, 37], large vision-language models [14, 15, 38], *etc.*). Recently, SigLIP [31] introduced pairwise sigmoid loss during training, enabling the visual encoder to demonstrate more advanced visual perception capabilities. To validate the compatibility of our *Dense Connector*, this paper conducted experiments on different visual encoders, including those of CLIP [8] and SigLIP [31].

### 2.2 Large Language Models

The exceptional text understanding and generation capabilities demonstrated by auto-regressive Large Language Models (LLMs) [39, 40, 41] have garnered significant attention. Subsequently, a plethora of LLMs [42, 11, 10, 43] have emerged, with notable open-source efforts like LLaMA [42] greatly propelling community contributions to LLMs research. Through instruction fine-tuning techniques, these models showcase human-like language interaction abilities, further further propelling advancements in natural language processing. Recent developments have seen LLMs scaled up or down to meet various application needs. Lightweight LLMs [44, 45, 20, 46] have been developed to address computational constraints, facilitating edge deployment. Conversely, in the pursuit of exploring the upper limits of LLMs, works such as [47, 19, 11, 20] have expanded LLM parameters, continuously pushing the boundaries of language capabilities. In this study, we validated the scalability of our *Dense Connector* by employing multiple LLMs ranging from 2.7B to 70B parameters.

### 2.3 Multimodal Large Language Models

After witnessing the success of LLMs, researchers have shifted their focus towards enabling LLMs to understand visual signals. To achieve this, prior research has proposed compressing visual embeddings using Q-former [14] into query embeddings, followed by transforming them into text embeddings through linear projection, or directly employing MLP projection [15] to connect the visual encoder with LLM. Furthermore, following the instruction tuning paradigm [48, 49], pioneering works [38, 15, 50] significantly boost the development of MLLMs through visual instruction tuning. Subsequently, by introducing larger-scale and higher-quality datasets, efforts such as [18, 25, 17, 24] have notably enhanced the visual understanding and reasoning capabilities of MLLMs. Additionally, there are works that introduce additional visual encoders [29, 30] or utilize higher-resolution images [25, 18, 24] to provide richer visual signal sources. Meanwhile, many studies [51, 52, 53] directly extend these above image-based methods to video conversational models by leveraging video instruction tuning datasets. In summary, these studies typically utilize high-level features from the frozen visual encoder as visual embeddings. However, we find that effectively leveraging offline features from different layers—the overlooked "free lunch"—can yield significant benefits. Then, we follow FreeVA [54] to directly extend the image model for video understanding without any additional video training.

## 3 Method

### 3.1 Overview

In Fig. 2(a), we illustrate the overall architecture of our model, using the mainstream LLaVA [15] framework as an example. It includes the pre-trained visual encoder $Vis(\cdot)$ and the Large Language Model $LLM(\cdot)$, alongside with our proposed *Dense Connector* $DC(\cdot)$. Similarly, our *DC* can be seamlessly extended to other high-resolution or dual-branch MLLMs, such as LLaVA-NeXT [25] and Mini-Gemini [18]. Formally, the introduction is as follows:

**Visual Encoder:** We utilize a CLIP pre-trained Vision Transformer (ViT) [21] as the visual encoder for extracting visual features. Initially, ViT partitions an image $X_i \in \mathbb{R}^{H \times W \times C}$ into a sequence of

 https://doi.org/10.52202/079017-1043

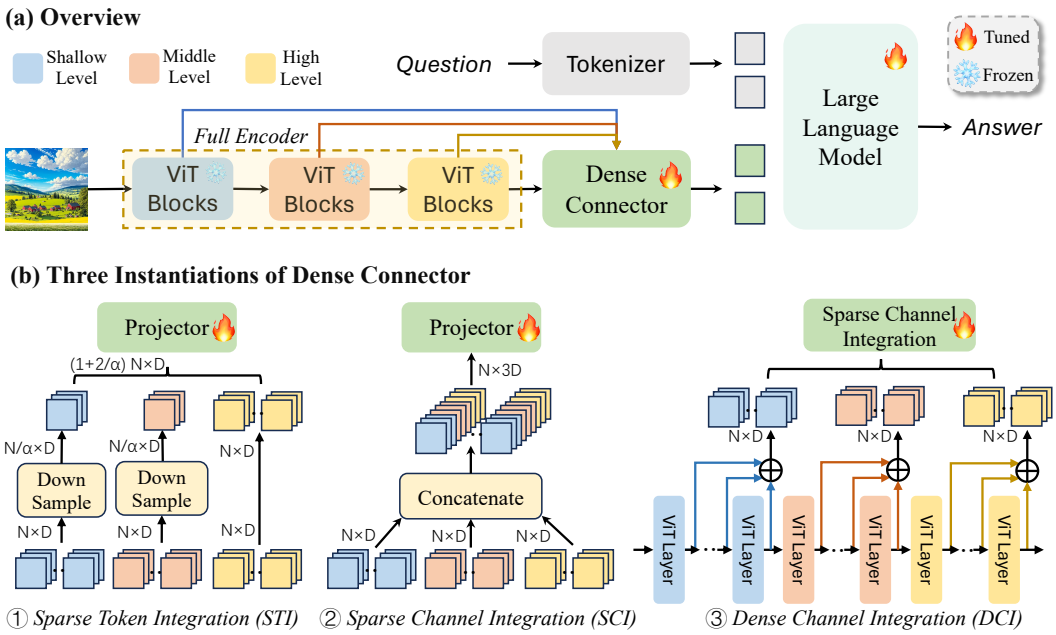



Figure 2: Dense Connector in MLLM: Overview and Three Instantiations. $N$ is the number of tokens, $D$ is the feature dimension, and $\alpha$ is the downsampling ratio.

non-overlapping patches. Each patch is then processed via convolution to produce visual tokens, which are subsequently input into ViT. This procedure yields $L$ layers of visual features $V \in \mathbb{R}^{L \times N \times D_v}$, where $N$ denotes the number of the visual tokens and $D_v$ denotes the feature dimension.

**Dense Connector:** The Dense Connector comprises two components: the first integrates multi-layer visual features, elaborated upon in detail in Sec. 3.2, while the second employs a learnable MLP to map the integrated visual features to the LLM's text space. The MLP consists of two linear layers with a GELU [55] activation function sandwiched between them. The first layer adjusts the visual hidden size $D_v$ to align with the LLM's hidden dimension $D_t$, while the second layer maintains the dimensionality at $D_t$. Upon processing through the Dense Connector, we acquire visual embeddings $e_v \in \mathbb{R}^{N \times D_t}$ that encapsulate information from multiple layers.

**Large Language Model:** The LLM processes textual data using a tokenizer and text embedding module to convert language into its input feature space. These text embeddings are concatenated with transformed visual embeddings before being fed into the LLM for subsequent predictions.

### 3.2 Dense Connector

Here we delve into three intuitive instantiations of the **Dense Connector**, each demonstrating superior performance compared to the baseline (*e.g.*, LLaVA-1.5 [16]). Among them, *Sparse Token Integration (STI)* and *Sparse Channel Integration (SCI)* sparsely select visual features from $K$ layers (indexed as $l_n$, where $1 \leq l_n < L$ and $1 \leq n \leq K$) spanning shallow, middle, and high levels out of the total $L$ layers of ViT, while *Dense Channel Integration (DCI)* utilizes features from all layers. These features are then fed into Dense Connector to generate visual embedding that can be "understood" by LLM.

**Sparse Token Integration (STI):** While existing methods typically rely solely on features from the final layer as the visual representation input for the LLM, our STI approach diverges from this by integrating features from multiple layers to enrich the visual input for the LLM. Recognizing that higher-level features contain richer semantic information crucial for visual signal perception in VLMs, we maintain the final layer features unchanged while downsampling additional visual features from other layers by using average pooling $avg(\cdot)$ with a stride $\alpha$. This downsampling reduces the number of visual tokens to $N' = N/\alpha$, mitigating computational overhead and redundancy. These visual features from various layers are concatenated along the token dimension and processed through a shared $MLP(\cdot)$, yielding more robust visual embedding $e_v \in \mathbb{R}^{(N+(k-1)\times N') \times D_t}$:

https://doi.org/10.52202/079017-1043

$$e_v = MLP(Concatenate([avg(V_{l_1}), ..., avg(V_{l_K}), V_L], dim = token)). \quad (1)$$

**Sparse Channel Integration (SCI):** We delve deeper into connecting multi-level features along the channel dimension. Subsequently, this feature is processed through an MLP projector to obtain visual embedding $e_v \in \mathbb{R}^{N \times D_t}$:

$$e_v = MLP(Concatenate([V_{l_1}, ..., V_{l_K}, V_L], dim = channel)). \quad (2)$$

The MLP projector serves dual functions: integrating various features as a fusion tool and facilitating the transformation of visual inputs into linguistic representations. This design ingeniously leverages the dimensionality scaling effect of the MLP projector, enabling the transformation of connected multi-layer features into the feature space of the LLM without requiring additional modules. Moreover, this method does not increase the number of tokens fed into the LLM, thereby avoiding any increase in the computational overhead of the LLM.

**Dense Channel Integration (DCI):** While *Sparse Channel Integration* incorporates features from $K$ layers, many visual feature from other layers remain unused. Concatenating all visual feature layers using STI or SCI leads to excessively high dimensions, posing challenges during training. To address these issues, we propose DCI, which builds upon the SCI method by integrating adjacent layers to reduce redundancy and high dimensionality. This approach ensures dense connectivity across a wider range of visual layers. Specifically, we partition the features of $L$ layers into $G$ groups, where each group comprises $M$ adjacent visual features, with $M = L/G$. Summing the features within each group, denoted as $GV_g$, finally yields $G$ fused visual representations:

$$GV_g = \frac{1}{M} \sum_{i=(g-1)M+1}^{gM} V_i, \quad 1 \leq g \leq G. \quad (3)$$

Subsequently, we concatenate these features from the $G$ groups with the final layer's features along the channel dimension before passing them through an MLP:

$$e_v = MLP(Concatenate([GV_1, ..., GV_G, V_L], dim = channel)). \quad (4)$$

### 3.3 Efficient Dense Connector for Visual Token Optimization

For MLLMs, each image is converted into hundreds or even thousands of visual tokens, and this large number of tokens increases the computational burden on autoregressive models. However, reducing the number of visual tokens generally leads to a noticeable drop in performance. In this paper, we leverage multi-layer visual features to compensate for the information loss caused by reducing visual tokens, enabling our method to achieve performance on par with LLaVA-v1.5 [16] while using several times fewer visual tokens, and surpassing other carefully designed efficient connectors [24, 38, 50, 56, 57] Specifically, after obtaining the visual embedding $e_v$ through the Dense Connector, we apply a parameter-free module, *i.e.* a 2D interpolation function, to downsample visual tokens. Then, these discrete visual tokens $e'_v$ are concatenated with the text tokens and fed into the LLM, resulting in a 3 times improvement in inference speed.

### 3.4 Training-Free Extension from Image to Video Conversational Models

Following FreeVA [54], we extend the image-based models trained as described above to video domain for video understanding. Specifically, given a video, we uniformly sample $T$ frames, and each frame is processed through the visual encoder and dense connector to obtain a visual embedding $e_v$. Consequently, we obtain an embedding sequence $\{e_{v_1}, ..., e_{v_T}\}$, which is then fed into the LLM.


https://doi.org/10.52202/079017-1043

# 4 Experiments

## 4.1 Implementation Details

**Architecture.** 1) Visual Encoders: To explore the generalization of the Dense Connector across different visual encoders, we select two mainstream options, namely CLIP-ViT-L-336px [8] and SigLIP-ViT-SO [31]. 2) LLMs: The Dense Connector is applied across various LLMs, spanning from 2.7B to 70B parameters. This includes Phi-2-2.7B [44], Vicuna-7B&13B [12], Hermes-2-Yi-34B [19], and Llama3-8B&70B-Instruct. 3) Dense Connector: For the 24-layer CLIP-ViT-L-336px [8], we specifically target visual features from the 8th, 16th, and final 24th layers for both STI and SCI. For STI, we apply a downsampling factor of $\alpha = 8$ for the features from the 8th and 16th layers. For DCI, we divide all layer features into two groups, each containing 12 layers (*i.e.*, 1-12, 13-24). Note that we use 1 to denote the stem layer of ViT, *e.g.*, 2-25 correspond to the 24 layers of ViT-L. Similarly, for the SigLIP ViT-SO [31], which has 27 layers, we partition the first 26 layers into two groups (*i.e.*, 1-13, 14-26).

**Training Datasets.** Data quality plays a crucial role in determining the performance of MLLMs. In this study, we examine the impact of two high-quality training datasets on our model: LLaVA-1.5 [16] and Mini-Gemini [18]. The LLaVA-1.5 pre-training dataset comprises 558K image captions, while its instruction tuning dataset contains 665K conversations. Mini-Gemini builds upon LLaVA-1.5, offering a larger dataset with 1.2M image-text caption pairs for alignment and 1.5M conversations for instruction tuning. Unless otherwise specified, all experimental results are based on the LLaVA-1.5 dataset to reduce training costs.

**Training Recipe.** We train all models on 8 NVDIA A100 GPUs with 40GB VRAM, except for the Hermes-2-Yi-34B and LLama-3-70B-Instruct, which utilize 32 NVDIA A100 GPUs with 80GB VRAM. Our training process comprises two stages: pre-training and instruction fine-tuning. In the pre-training phase, we initialize the visual encoder and LLM with pre-trained weights, while the Dense Connector is randomly initialized. Here, we freeze the visual encoder and the LLM, updating only the parameters of the Dense Connector. The model undergoes pre-training for one epoch with a global batch size of 256 and a learning rate of 1e-3. Subsequently, in the instruction fine-tuning stage, we maintain the visual encoder frozen while updating the Dense Connector and the LLM. Fine-tuning is performed for 1 epoch with a global batch size of 128 and a learning rate of 2e-5. For models using LoRA fine-tuning, we set the LoRA rank to 128 and LoRA alpha to 256. When scaling up the LLM to larger parameter sizes, such as LLama-3-70B-Instruct, we apply LoRA fine-tuning due to memory constraints. In this setup, we set the LoRA rank to 128 and LoRA alpha to 256.

**Evaluation.** We present comprehensive results across various image and video evaluation benchmarks. For image datasets, we include GQA [58], VQAV2 (VQA$^{v2}$) [59], ScienceQA (SQA$^{I}$) [60], TextVQA (VQA$^{T}$) [61], POPE [62], MathVista (Math) [63], MMBench (MMB) [64], MM-Vet (MMV) [65], MMMU [66], LLaVA-Bench-In-the-Wild (LBW) [15], and MME [67]. Additionally, we evaluate zero-shot performance on open-ended video question-answering benchmarks such as MSVD-QA [68], ActivityNet-QA [69], MSRVTT-QA [70], and the newly proposed generative performance benchmark [51], include evaluation metrics such as Correctness of Information (CI), Detail Orientation (DO), Contextual Understanding (CU), Temporal Understanding (TU), and Consistency (CO).

## 4.2 Ablation Study

**Study on Instantiations of Dense Connector.** In Tab. 1, we discuss three proposed methods of Dense Connector. 1) **Sparse Token Integration (STI):** This method uses visual features from different hierarchical levels as independent visual prompts for the LLM, allowing it to perceive a more diverse set of visual features. We select features from the 8th, 16th, and 24th layers, resulting in significant improvements across various datasets, particularly achieving a 2.9% increase on the MMB [64]. Further expanding the selection to include the 8th, 16th, 20th, and 24th layers enhances performance but also increases token count, leading to higher training costs and inference time. 2) **Sparse Channel Integration (SCI):** This method efficiently uses the MLP projector for feature fusion and projection, integrating multiple layers of visual features into visual embeddings. SCI boosts performance with minimal additional computational cost, achieving peak performance with features from the 8th, 16th, and 24th layers, resulting in a 1.7% improvement on GQA. SCI performs better than STI and mitigates the computational costs associated with higher token counts. However, expanding the range of visual

Table 1: Ablations on Visual Layer Selection in Dense Connector. Here, we explore three instantiations (*STI*, *SCI*, and *DCI*) of our Dense Connector integrated with the baseline (*i.e.*, LLaVA-1.5 [16]), which utilizes a 24-layer CLIP-ViT-L-336px.

| Model | Layer Index | **GQA** | **VQA**$^{v2}$ | **SQA**$^{I}$ | **VQA**$^{T}$ | **POPE** | **MMB** | **MMV** | **LBW** |
|---|---|---|---|---|---|---|---|---|---|
| Baseline | 24 | 62.0 | 78.5 | 66.8 | 58.2 | 85.9 | 64.3 | 31.1 | 65.4 |
| + *STI* | 8,16,24 | 63.3 | 79.1 | 68.0 | 58.0 | 85.8 | 67.2 | 30.9 | 65.5 |
| + *STI* | 8,16,20,24 | 63.0 | 79.1 | 68.0 | 58.8 | 85.9 | 67.6 | 30.8 | 65.7 |
| + *SCI* | 8,16,24 | 63.7 | 79.2 | 68.9 | 58.2 | 86.1 | 66.2 | 32.2 | 66.0 |
| + *SCI* | 16,24 | 63.0 | 79.0 | 67.6 | 58.2 | 86.0 | 65.6 | 31.7 | 65.6 |
| + *SCI* | 8,16,20,24 | 63.6 | 79.2 | 67.0 | 58.1 | 86.0 | 65.8 | 31.9 | 66.0 |
| + *DCI* | (1-8),(9-16),(17-24) | 63.6 | 79.3 | 67.8 | 58.6 | 86.3 | 66.5 | 32.6 | 66.0 |
| + *DCI* | (1-12),(13-24) | **63.8**$^{1.8\uparrow}$ | **79.5**$^{1.0\uparrow}$ | **69.5**$^{2.7\uparrow}$ | **59.2**$^{1.0\uparrow}$ | **86.6**$^{0.7\uparrow}$ | **66.8**$^{2.5\uparrow}$ | **32.7**$^{1.6\uparrow}$ | **66.1**$^{0.7\uparrow}$ |

Table 2: Exploring the Compatibility and Scalability of Dense Connector (DC). Scaling results on visual encoder (VE), resolution (Res.), pre-training (PT) / instruction tuning (IT) data, and LLM are provided. "0.5M+0.6M" denotes the training data from LLaVA-1.5 [16], while "1.2M+1.5M" denotes the data from Mini-Gemini [18]. * indicates results evaluated using official model.

| Method | VE | Res. | PT+IT | LLM | **GQA** | **SQA**$^{I}$ | **VQA**$^{T}$ | **MMB** | **MMV** | **MMMU**$^{v}$ | **Math** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *Scaling to more powerful visual encoder* | | | | | | | | | | | |
| LLaVA [16] | CLIP-L | 336 | 0.5M+0.6M | Vicuna-7B | 62.0 | 66.8 | 58.2 | 64.3 | 31.1 | 35.3* | 24.9* |
| LLaVA [16] | CLIP-L | 336 | 0.5M+0.6M | Vicuna-13B | 63.3 | 71.6 | 61.3 | 67.6 | 36.1 | 36.4 | 27.6 |
| DC (*w/* LLaVA) | CLIP-L | 336 | 0.5M+0.6M | Vicuna-7B | 63.8 | 69.5 | 59.2 | 66.8 | 32.7 | 34.8 | 26.9 |
| DC (*w/* LLaVA) | SigLIP-SO | 384 | 0.5M+0.6M | Vicuna-7B | 64.2 | 70.5 | 62.6 | 68.4 | 35.4 | **36.7** | 25.5 |
| DC (*w/* LLaVA) | SigLIP-SO | 384 | 0.5M+0.6M | Vicuna-13B | **65.4** | **73.0** | **64.7** | **71.4** | **41.6** | 34.3 | **29.6** |
| *Scaling to larger-scale training data* | | | | | | | | | | | |
| DC (*w/* LLaVA) | SigLIP-SO | 384 | 1.2M+1.5M | Vicuna-7B | 63.8 | 72.9 | 64.6 | 71.7 | 45.0 | 35.8 | 33.1 |
| DC (*w/* LLaVA) | SigLIP-SO | 384 | 1.2M+1.5M | Vicuna-13B | **64.6** | **77.1** | **65.0** | **74.4** | **47.7** | **37.2** | **36.5** |
| *Scaling to high resolution with a dual visual encoder* | | | | | | | | | | | |
| MGM [18] | CLIP-L +ConvX-L | 336 +768 | 1.2M+1.5M | Vicuna-7B | 62.6* | 70.4* | 65.2 | 69.3 | 40.8 | 36.1 | 31.4 |
| MGM [18] | CLIP-L +ConvX-L | 336 +768 | 1.2M+1.5M | Vicuna-13B | 63.4* | 72.6* | 65.9 | 68.5 | 46.0 | 38.1 | 37.0 |
| DC (*w/* MGM) | CLIP-L +ConvX-L | 336 +768 | 1.2M+1.5M | Vicuna-7B | 63.3 | 70.7 | 66.0 | 70.7 | 42.2 | 36.8 | 32.5 |
| DC (*w/* MGM) | CLIP-L +ConvX-L | 336 +768 | 1.2M+1.5M | Vicuna-13B | **64.2** | **74.9** | **66.7** | **70.7** | **49.8** | **39.3** | **38.1** |
| *Scaling to dynamic high resolution* | | | | | | | | | | | |
| LLaVA-NeXT [16] | CLIP-L | AnyRes | 0.5M+0.6M | Vicuna-7B | 64.0 | 69.5 | 64.5 | 66.5 | 33.1 | 35.4 | 25.7 |
| DC (*w/* LLaVA) | CLIP-L | AnyRes | 0.5M+0.6M | Vicuna-7B | 64.6 | **70.5** | 65.6 | **67.4** | 33.7 | **37.6** | 26.2 |
| DC (*w/* LLaVA) | SigLIP-SO | AnyRes | 0.5M+0.6M | Vicuna-7B | **64.8** | 69.3 | **66.5** | 67.2 | **34.8** | 36.3 | **27.0** |

layers within SCI does not yield additional performance gains, suggesting that merely extending the range of visual feature layers is ineffective. 3) **Dense Channel Integration (DCI):** Building on the performance enhancements of SCI, DCI integrates a broader array of visual features using grouped additive fusion to produce robust visual representations. We divide the visual features into 2 or 3 groups, each with an equal number of layers. For the CLIP-L model, each group combines features from 12 or 8 layers, respectively. Splitting them into 2 groups demonstrates superior performance, achieving improvements of 2.7% on SQA [60] and 2.5% on MMB [64] compared to the baseline. The experimental results illustrate that utilizing multi-layer visual features enhances the visual perception capabilities of the MLLMs, leading to more accurate responses. Unless otherwise specified, we employ DCI as the default instantiation of the Dense Connector for optimal performance.

**Study on Visual Encoders and Training Dataset Impacts.** Given our method's reliance on multi-layer visual features, it is crucial to assess its impact across various visual backbones. As shown in Tab. 2, we first replace the CLIP-ViT-L [8] with the more advanced visual encoder SigLIP-ViT-


https://doi.org/10.52202/079017-1043

Table 3: Comparison of Efficient Dense Connector with Other Efficient Methods. * indicates results evaluated using official model.

| Method | Res. | #Token | PT+IT | LLM | **GQA** | **VQA$^{v2}$** | **SQA$^{I}$** | **VQA$^{T}$** | **MMB** | **MMV** | **Math** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LLaVA [16] | 336 | 576 | 0.5M+0.6M | Vicuna-7B | 62.0 | 78.5 | 66.8 | **58.2** | 64.3 | 31.1 | 24.9* |
| Qwen-VL-Chat [24] | 448 | 256 | 1.4B+50M | Qwen-7B | 57.5 | 68.2 | 61.5 | - | - | - | - |
| TokenPacker [56] | 336 | 144 | 0.5M+0.6M | Vicuna-7B | 61.9 | 77.9 | - | - | 65.1 | 33.0 | - |
| Dense Connector | 336 | 144 | 0.5M+0.6M | Vicuna-7B | **62.8** | **79.4** | **68.8** | 58.1 | **67.6** | **34.4** | **25.8** |

SO [31]. Leveraging the enhanced multi-layer visual features from SigLIP-ViT-SO, our Dense Connector demonstrates further performance improvements. Additionally, we investigate the influence of training datasets on the effectiveness of the Dense Connector. By fine-tuning our model using the larger dataset [18], we observe notable performance gains across most benchmark evaluations. The results indicate that more training data significantly enhance model performance. Specifically, our model with Vicuna-13B achieves accuracy of 36.5% on MathVista [63] and 77.1% on SQA [60], underscoring the significant benefits of increased training data. Moreover, when using the same training data and the LLM (*i.e.*, Vicuna-7B), our Dense Connector surpasses the dual encoder structure of Mini-Gemini [18] across the majority of benchmarks. Specifically, it achieves performance gains of 2.4% on the MMB [64], 4.2% on MM-Vet [65], and 1.7% on the MathVista [63] benchmark.

**Study on High-Resolution Setting.** The use of high-resolution images to enhance detail representation in MLLMs has garnered considerable attention [25, 18, 26, 27, 28]. In this paper, we extend Dense Connector to the Mini-Gemini (MGM) [18] and LLaVA-NeXT [25], showcasing its plug-and-play capability. For MGM, we keep the high-resolution features from ConvNeXT [71] intact, applying DCI exclusively to the CLIP features. We observe significant improvements across various benchmarks, as detailed in Tab. 2, including MathVista [63], MMB [64], and MM-Vet [65], with enhancements of 1.1%, 2.2%, and 3.8%, respectively. In addition to the high-resolution architecture of the dual visual encoder, we also extend the Dense Connector to dynamic high-resolution, specifically using the AnyRes technology from LLaVA-NeXT [25]. For a fair comparison, we provide a baseline for LLaVA-NeXT trained on the same dataset. As shown in Tab. 2, Dense Connector achieves overall improvements compared to the dynamic resolution method LLaVA-NeXT as well.

**Study on Efficient Dense Connector.** To achieve faster inference speed, we investigate an efficient Dense Connector in this study, which can accelerate inference by 3 times. As described in Sec. 3.3, we use a 2D bilinear interpolation function to downsample the visual tokens by a factor of 4, reducing the number of tokens from 576 to 144, which decreases the training time during in the second stage on 8 A100 GPUs from 9 hours to 6.5 hours. With the same configuration of using 144 visual tokens, Dense Connector outperforms the carefully designed efficient connector method Tokenpacker [56] by 0.9%, 1.5%, 2.5% and 1.4% on GQA [58], VQAv2 [59], MMB [64], and MM-Vet [65], respectively.

### 4.3 Main Results

**Comparison with SoTAs in Image Understanding.** In Tab. 4, we scale the LLMs from 2.7B to 70B parameters and compare them with state-of-the-art MLLMs. When considering lightweight models, our Dense Connector surpasses the previous MLLM, TinyLlava [72], achieving a 1.7% enhancement on the MM-Vet benchmark using the same fine-tuning data and foundation model. Furthermore, using same training data and LLM, our Dense Connector outperforms the LLaVA-1.5 Vicuna 13B [16] with substantial gains of 2.1%, 3.7%, and 5.5% on the GQA [58], MMB [64], and MM-Vet [65] benchmarks, respectively. Notably, even with data solely from LLaVA-1.5, our 13B model achieves performance comparable to MGM [18], which is trained on larger datasets, including 1.2M+1.5M data. Moreover, utilizing the advanced open-source LLM Llama3-8B-Instruct, our model significantly surpasses LLaVA-LLama3 [76] with improvements of 5.5%, and 52 on MMB [64], and MME$^{p}$ [67], respectively, highlighting the contribution of our Dense Connector. By scaling up the LLM to 34B and 70B, Dense Connector achieves further improvements leveraging more powerful language models. The 70B model attains scores of 82.4% on SQA [60] and 79.4% on MMBench [64]. We then increase the resolution using AnyRes technology [25] and fully fine-tuned the LLM. Our 13B model outperforms MGM and LLaVA-NeXT on MMBench [64] and SQA [60], achieving scores of 72.3% and 72.6%. The 34B model achieves scores of 81.2%, 59.2%, and 97.7% on MMBench [64], MM-Vet [65], and LLaVA-Bench-in-the-Wild [16], respectively.

Table 4: Comparisons with State-of-the-Arts. * indicates the dataset have been used for training, and † indicates the dataset is not publicly accessible. "PT," "IT," and "Res." denote pre-training data, instruction fine-tuning data, and image resolution, respectively.

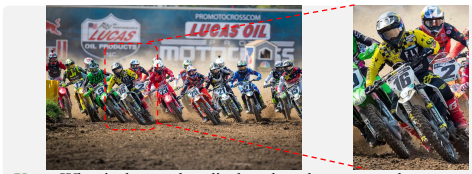
| Method | PT+IT | Res. | LLM | SQA$^I$ | MMB | MME$^P$ | MM-Vet | MMMU$^v$ | Math | LLaVA$^W$ | GQA |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MobileVLM V2 [57] | 1.2M+3.6M | 336 | ML-2.7B | 70.0 | 63.2 | 1441 | – | – | – | – | 61.1 |
| TinyLLaVA [72] | 0.5M+0.6M | 384 | Phi2-2.7B | 69.9 | – | – | 32.1 | – | – | 67.9 | 61.3 |
| mPLUG-Owl2 [73] | 348M+1.2M | 448 | Llama2-7B | 68.7 | 64.5 | 1450 | 36.2 | 32.7 | 22.2 | – | 56.1 |
| Qwen-VL-Chat† [24] | 1.4B+50M | 448 | Qwen-7B | 68.2 | 60.6 | 1488 | – | – | – | – | 57.5* |
| LLaVA-v1.5 [16] | 0.5M+0.6M | 336 | Vicuna-13B | 71.6 | 67.7 | 1531 | 36.1 | 36.4 | 27.6 | 72.5 | 63.3 |
| ShareGPT4V [17] | 1.2M+0.7M | 336 | Vicuna-13B | 71.2 | 68.5 | 1619 | 43.1 | – | – | 79.9 | 64.8 |
| MobileVLM V2 [57] | 1.2M+3.6M | 336 | Vicuna-7B | 74.8 | 70.8 | 1559 | – | – | – | – | 64.6 |
| LLaMA-VID [74] | 0.8M+0.7M | 336 | Vicuna-7B | 70.0 | 66.6 | 1542 | – | – | – | – | 65.0* |
| SPHINX-Plus [75] | 16M | 448 | Llama2-13B | 74.2 | 71.0 | 1458 | 47.9 | – | 36.8 | 71.7 | – |
| LLaVA-LLaMA3 [76] | 0.5M+0.6M | 336 | Llama3-8B | 73.3 | 68.9 | 1506 | – | 36.8 | – | – | 63.5 |
| CuMo [77] | 0.5M+0.6M | 336 | Mistral-7B | 71.7 | 69.6 | 1429 | 34.3 | – | – | 68.8 | 63.2 |
| MM1 [77] | 3B+1.4M | 1344 | MM1-7B | 72.6 | 79.0 | 1529 | 42.1 | 37.0 | 35.9 | 81.5 | – |
| VILA [78] | 50M+1M | 336 | Llama-2-13B | 73.7 | 70.3 | 1570 | 38.8 | – | – | 73.0 | 63.3* |
| Mini-Gemini [18] | 1.2M+1.5M | 336+768 | Vicuna-13B | 72.6 | 68.5 | 1565 | 46.0 | 38.1 | 37.0 | 87.7 | 63.4 |
| LLaVA-NeXT [25] | 0.5M+0.7M | 336$_{AnyRes}$ | Vicuna-13B | 73.6 | 70.0 | 1575 | 48.4 | 36.2 | 35.3 | 87.3 | 65.4 |
| *Scaling to a wider range of parameter sizes (2B → 70B) for LLMs* | | | | | | | | | | | |
| Dense Connector | 0.5M+0.6M | 384 | Phi2-2.7B | 70.3 | 70.5 | 1487 | 33.8 | 36.6 | 28.2 | 65.1 | 61.5 |
| Dense Connector | 0.5M+0.6M | 384 | Vicuna-7B | 70.5 | 68.4 | 1523 | 35.4 | 36.7 | 25.5 | 67.4 | 64.4 |
| Dense Connector | 0.5M+0.6M | 384 | Vicuna-13B | 73.0 | 71.4 | 1569 | 41.6 | 34.3 | 29.6 | 73.6 | **65.4** |
| Dense Connector | 0.5M+0.6M | 384 | Llama3-8B | 75.2 | 74.4 | 1558 | 34.6 | 40.4 | 28.6 | 68.8 | 65.1 |
| Dense Connector | 0.5M+0.6M | 384 | Yi-34B$_{LoRA}$ | 80.5 | 77.7 | 1588 | 41.0 | 47.1 | 33.5 | 75.1 | 63.9 |
| Dense Connector | 0.5M+0.6M | 384 | Llama3-70B$_{LoRA}$ | **82.4** | 79.4 | 1622 | 46.1 | 47.0 | 32.9 | 74.5 | 64.0 |
| Dense Connector | 1.2M+1.5M | 384 | Vicuna-13B | 77.1 | 74.4 | 1579 | 47.8 | 37.2 | 36.5 | 88.9 | 64.6 |
| Dense Connector | 1.2M+1.5M | 384$_{AnyRes}$ | Vicuna-7B | 72.0 | 69.2 | 1535 | 44.4 | 36.4 | 32.7 | 88.8 | 63.9 |
| Dense Connector | 1.2M+1.5M | 384$_{AnyRes}$ | Vicuna-13B | 75.2 | 72.3 | 1573 | 47.0 | 36.8 | 35.5 | 93.2 | 64.3 |
| Dense Connector | 1.2M+1.5M | 384$_{AnyRes}$ | Yi-34B | 78.0 | **81.2** | **1696** | **59.2** | **51.8** | **40.0** | **97.7** | **66.6** |

Table 5: Comparisons with Leading Methods on Zero-shot Video QA Benchmarks. Following FreeVA [54], we specify the GPT-3.5 versions used for evaluation to ensure fairness in performance comparison across different versions. "MAR" denotes the `GPT-3.5-Turbo-0301`, "JUN" denotes the `GPT-3.5-Turbo-0613`, and "JAN" denotes the latest `GPT-3.5-Turbo-0125`.

| Method | LLM Size | GPT-3.5 Version | Train Free | MSVD-QA | | MSRVTT-QA | | ActivityNet-QA | | Video-ChatGPT benchmark | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Acc | Score | Acc | Score | Acc | Score | CI | DO | CU | TU | CO |
| FrozenBiLM [79] | 0.9B | MAR | ✗ | 33.8 | – | 16.7 | – | 25.9 | – | – | – | – | – | – |
| Video-LLaMA[52] | 7B | MAR | ✗ | 51.6 | 2.5 | 29.6 | 1.8 | 12.4 | 1.1 | 1.96 | 2.18 | 2.16 | 1.82 | 1.79 |
| LLaMA-Adapter [80] | 7B | MAR | ✗ | – | – | – | – | – | – | 2.03 | 2.32 | 2.30 | 1.98 | 2.15 |
| VideoChat [81] | 7B | MAR | ✗ | 56.3 | 2.8 | 45.0 | 2.5 | 26.5 | 2.2 | 2.23 | 2.50 | 2.53 | 1.94 | 2.24 |
| Video-ChatGPT [51] | 7B | MAR | ✗ | 64.9 | 3.3 | 49.3 | 2.8 | 35.2 | 2.7 | 2.50 | 2.57 | 2.69 | 2.16 | 2.20 |
| VaQuitA [82] | 7B | MAR | ✗ | 74.6 | 3.7 | 68.6 | 3.3 | 48.8 | 3.3 | – | – | – | – | – |
| LLaVA+FreeVA [54] | 7B | MAR | ✓ | 81.5 | 4.0 | 72.9 | 3.5 | 58.3 | 3.5 | 2.88 | 2.52 | 3.25 | 2.32 | 3.07 |
| BT-Adapter [83] | 7B | JUN | ✗ | 67.5 | 3.7 | 57.0 | 3.2 | 45.7 | 3.2 | 2.68 | 2.69 | 3.27 | 2.34 | 2.46 |
| Video-LLaVA [53] | 7B | JUN | ✗ | 70.7 | 3.9 | 59.2 | 3.5 | 45.3 | 3.3 | – | – | – | – | – |
| LLaMA-VID [74] | 13B | JUN | ✗ | 70.0 | 3.7 | 58.9 | 3.3 | 47.5 | 3.3 | 3.07 | 3.05 | 3.60 | 2.58 | 2.63 |
| LLaVA+FreeVA [54] | 13B | JUN | ✓ | 71.8 | 3.8 | 59.2 | 3.3 | 54.5 | 3.5 | 2.90 | 2.52 | 3.26 | 2.32 | 3.07 |
| LLaVA+FreeVA [54] | 13B | JAN | ✓ | 74.4 | 4.1 | 61.1 | 3.6 | 51.6 | 3.5 | 2.88 | 2.52 | 3.25 | 2.34 | 3.05 |
| DC+FreeVA | 7B | JAN | ✓ | 75.0 | 4.1 | 58.4 | 3.5 | 52.2 | 3.5 | 2.80 | 2.51 | 3.17 | 2.22 | 3.05 |
| DC+FreeVA | 13B | JAN | ✓ | 75.1 | 4.1 | 60.8 | 3.5 | 52.6 | 3.5 | 2.85 | 2.53 | 3.23 | 2.29 | 2.96 |
| DC+FreeVA | 34B | JAN | ✓ | 77.4 | 4.2 | 62.1 | 3.6 | 55.8 | 3.6 | 3.00 | 2.53 | 3.25 | 2.65 | 2.92 |

**Comparison with SoTAs in Video Understanding.** Building on the training-free paradigm of FreeVA [54] for image-to-video adaptation, we directly apply our models, originally trained on image-text datasets, to video dialogues. As indicated in Tab. 5, the visual enhancement capabilities of the Dense Connector significantly enhance the video comprehension of our 13B model. This model surpasses the baseline LLaVA-1.5 [16] with FreeVA [54] on MSVD [68] and ActivityNet [69]. By scaling up the LLM to 34B, our model demonstrates a marked improvement in video understanding capabilities, achieving state-of-the-art performance on video evaluation benchmarks with accuracies of 77.4% on MSVD [68], 62.1% on MSR-VTT [70], and 55.8% on ActivityNet [69].

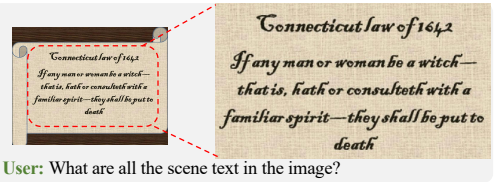

https://doi.org/10.52202/079017-1043



**User:** What is the number displayed on the motorcycle ridden by the motorcyclist wearing a yellow suit?

**DC:** The motorcyclist wearing a yellow suit is riding a motorcycle with the number **16** on it.

(a) Detail Reasoning




**User:** What are all the scene text in the image?

**DC:** The scene text in the image reads: "Connecticut Law of 1642. If any man or woman be a witch, that is hath or consulteth with a familiar spirit, they shall be put to death."

(b) OCR

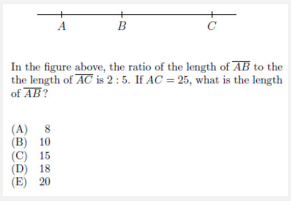

**User:** What is the average total fueling cost excluding the Ford F150?

**DC:** To calculate the average total fueling cost excluding the Ford F150, we would add the costs of the Toyota RAV4 and the Honda Civic and then divide by two. The Toyota RAV4's cost is $82.56 and the Honda Civic's cost is $70.55. Adding these two amounts gives us $153.11. Dividing this sum by two gives us an average of **$76.55**.

**User:** What is the highest fueling cost?

**DC:** The highest fueling cost, as shown on the bar chart, is for the Ford F-150, with a total fueling cost of **$130.96**.

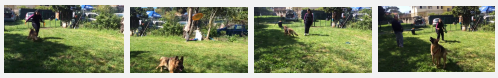
(c) Table Reading




**User:** Answer the question in the figure and provide the options.

**DC:** To find the length of AB, we can use the given ratio of the lengths of AB to AC. The ratio is 2:5, which means that for every 2 units of AB, there are 5 units of AC. Since AC is 25 units, we can set up a proportion to find the length of AB:
2/5 = AB/25
To solve for AB, we can cross-multiply:
2 × 25 = 5 × AB, 50 = 5 × AB
Dividing both sides by 5 gives us:
AB = 50/5, AB = 10
Therefore, the length of AB is 10 units.
The correct answer is **B) 10.**

(d) Math Solving



**User:** What are they doing in the video?

**DC:** The people in the video are playing with their dogs in a grassy field. They are **throwing a frisbee** for the dogs to catch, and the dogs are running to catch it.

**User:** What breed is this dog?

**DC:** The dog in the video is a **German Shepherd**.

(e) Video Understanding

**User:** What animal is in the video, and what is it doing?

**DC:** The animal in the video is a giraffe, and it is standing on a diving board, preparing to **jump into the pool**.

**User:** Is this video real or synthesized?

**DC:** This is a **computer-generated video** of a giraffe diving into a pool, which is not real.

(f) Video Reasoning

Figure 3: Quantitative Results for Image and Video dialogues. Figures (a) through (d) pertain to image understanding, while figures (e) and (f) relate to video understanding.

**Qualitative Results.** In Fig. 3, we illustrate our model's exceptional visual understanding and text generation capabilities across various scenarios, encompassing both image and video. More qualitative results are provided in Appendix A.5.

## 5 Conclusion and Limitation

In this paper, we introduce the Dense Connector, a novel plug-and-play module that enhances visual perception capabilities of MLLMs by densely integrating multi-layer visual features. We instantiated three types of Dense Connector and validate the efficacy of it across a diverse array of vision encoders, LLMs, and training datasets, demonstrating substantial improvements in performance across multiple evaluation benchmarks. Dense Connector can be easily integrated into existing MLLMs. In this work, we incorporate the Dense Connector into mainstream model LLaVA and high-resolution method Mini-Gemini, demonstrating its versatility and generalization capabilities.

**Limitations:** Our three Dense Connector instantiations do not introduce additional parameters, leaving room for further exploration. We have not yet found an effective method for incorporating additional parameters. Future research will focus on discovering more efficient ways to connect visual and language models for better modality alignment.

https://doi.org/10.52202/079017-1043

## References


[1] OpenAI. Chatgpt. `https://openai.com/blog/chatgpt/`, 2023.

[2] OpenAI. Gpt-4v(ision) system card. 2023.

[3] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[4] Wenhao Wu, Huanjin Yao, Mengxi Zhang, Yuxin Song, Wanli Ouyang, and Jingdong Wang. Gpt4vis: What can gpt-4 do for zero-shot visual recognition? *arXiv preprint arXiv:2311.15732*, 2023.

[5] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023.

[6] Chaoyi Wu, Jiayu Lei, Qiaoyu Zheng, Weike Zhao, Weixiong Lin, Xiaoman Zhang, Xiao Zhou, Ziheng Zhao, Ya Zhang, Yanfeng Wang, et al. Can gpt-4v (ision) serve medical applications? case studies on gpt-4v for multimodal medical diagnosis. *arXiv preprint arXiv:2310.09909*, 2023.

[7] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration. *arXiv preprint arXiv:2311.12015*, 2023.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

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

[10] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

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

[12] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[13] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[14] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

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

[16] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[17] Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023.

[18] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024.

[19] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*, 2024.


 https://doi.org/10.52202/079017-1043

[20] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[24] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[25] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[26] Bo Li, Peiyuan Zhang, Jingkang Yang, Yuanhan Zhang, Fanyi Pu, and Ziwei Liu. Otterhd: A high-resolution multi-modality model. *arXiv preprint arXiv:2311.04219*, 2023.

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

[28] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[29] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Xiaopeng Zhang, Jin Li, Hongkai Xiong, and Qi Tian. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023.

[30] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024.

[31] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[32] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[33] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. In *ICLR*, 2021.

[34] Bo Fang, Wenhao Wu, Chang Liu, Yu Zhou, Yuxin Song, Weiping Wang, Xiangbo Shu, Xiangyang Ji, and Jingdong Wang. Uatvr: Uncertainty-adaptive text-video retrieval. In *ICCV*, 2023.

[35] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10704–10713, 2023.

[36] Wenhao Wu, Zhun Sun, Yuxin Song, Jingdong Wang, and Wanli Ouyang. Transferring vision-language models for visual recognition: A classifier perspective. *International Journal of Computer Vision*, pages 1–18, 2023.

[37] Wenhao Wu, Xiaohan Wang, Haipeng Luo, Jingdong Wang, Yi Yang, and Wanli Ouyang. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In *CVPR*, pages 6620–6630, 2023.

[38] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[39] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[40] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[41] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[42] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[43] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[44] Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 2023.

[45] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

[46] Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, et al. Stable lm 2 1.6 b technical report. *arXiv preprint arXiv:2402.17834*, 2024.

[47] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[48] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[49] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[50] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[51] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023.

[52] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[53] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[54] Wenhao Wu. Freeva: Offline mllm as training-free video assistant. *arXiv preprint arXiv:2405.07798*, 2024.

[55] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[56] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *arXiv preprint arXiv:2407.02392*, 2024.

[57] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*, 2024.

 https://doi.org/10.52202/079017-1043

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

[59] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[60] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

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

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

[63] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

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

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

[66] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.

[67] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[68] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011.

[69] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[70] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[71] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[72] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024.

[73] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv preprint arXiv:2311.04257*, 2023.

[74] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023.

[75] Peng Gao, Renrui Zhang, Chris Liu, Longtian Qiu, Siyuan Huang, Weifeng Lin, Shitian Zhao, Shijie Geng, Ziyi Lin, Peng Jin, et al. Sphinx-x: Scaling data and parameters for a family of multi-modal large language models. *arXiv preprint arXiv:2402.05935*, 2024.

[76] XTuner Contributors. Xtuner: A toolkit for efficiently fine-tuning llm. `https://github.com/InternLM/xtuner`, 2023.

[77] Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. Cumo: Scaling multimodal llm with co-upcycled mixture-of-experts. *arXiv preprint arXiv:2405.05949*, 2024.

[78] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. *arXiv preprint arXiv:2312.07533*, 2023.

[79] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 35:124–141, 2022.

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

[81] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[82] Yizhou Wang, Ruiyi Zhang, Haoliang Wang, Uttaran Bhattacharya, Yun Fu, and Gang Wu. Vaquita: Enhancing alignment in llm-assisted video understanding. *arXiv preprint arXiv:2312.02310*, 2023.

[83] Ruyang Liu, Chen Li, Yixiao Ge, Ying Shan, Thomas H Li, and Ge Li. One for all: Video conversation is feasible without video instruction tuning. *arXiv preprint arXiv:2309.15785*, 2023.

[84] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[85] Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang, Ziwei Liu, and Chunyuan Li. Llava-next: Stronger llms supercharge multimodal capabilities in the wild, May 2024.

[86] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

 https://doi.org/10.52202/079017-1043

# A Appendix

## A.1 Further Exploration and Analysis of the Dense Connector

In Sec. 3.2, we provide a detailed discussion of three instantiation methods of the Dense Connector: Sparse Token Integration (STI), Sparse Channel Integration (SCI), and Dense Channel Integration (DCI). Applying our proposed Dense Connector to representative VLMs (*e.g.*, LLaVA [16]) results in notable performance improvements. In this section, building on the STI, SCI, and DCI methods, we explore instantiations of Dense Connectors with additional learnable parameters.

**Sparse Token Integration with 1D Convolutional Downsampling.** The STI instantiation method concatenates visual tokens from various layers, which significantly increases the total number of tokens. To reduce redundancy and computational overhead, we initially employ average pooling to reduce the number of tokens from shallow layer features. In computer vision, using convolution for downsampling is a common practice [84]. Here, we replace average pooling with a single 1D convolutional layer $Conv_{1D}(\cdot)$, to downsample the shallow layer tokens. We set both the kernel size and stride to 8, maintaining consistency with the average pooling as detailed in Sec. 4.1. The formula is as follows:

$$e_v = MLP(Concatenate([Conv_{1D}(V_{l_1}), ..., Conv_{1D}(V_{l_K}), V_L], dim = token)). \tag{5}$$

**Sparse Channel Integration with 2D Convolutional Modelling.** The SCI method utilizes the projector as both a feature fusion tool and a modality mapper, transforming multi-layer visual features concatenated along the channel dimension into the input feature space of the LLMs. By straightforwardly concatenating multi-layer visual features, the SCI method achieves significant performance enhancements for VLMs with minimal computational overhead. As a central element of the Dense Connector, we aim to enhance local perception abilities by processing visual features through a 3×3 2D convolution $Conv_{2D}$, with shared weights prior to feature concatenation:

$$e_v = MLP(Concatenate([Conv_{2D}(V_{l_1}), ..., Conv_{2D}(V_{l_K}), Conv_{2D}(V_L)], dim = channel)). \tag{6}$$

**Dense Channel Integration with Linear Layer.** The DCI method initially groups visual features and then aggregates them, enabling the fusion of adjacent visual features without expanding the dimensionality. This technique mitigates the issues of feature redundancy and excessive channel dimensions that arise in the SCI method from incorporating too many layers. To enhance the integration of features from different visual layers across groups, each layer of visual features is processed through a linear layer with shared weights. The formula is as follows:

$$GV_g = \frac{1}{M} \sum_{i=(g-1)M+1}^{gM} Linear(Ln(V_i)), \quad 1 \leq g \leq G. \tag{7}$$

$$e_v = MLP(Concatenate([GV_1, ..., GV_G, V_L], dim = channel)), \tag{8}$$

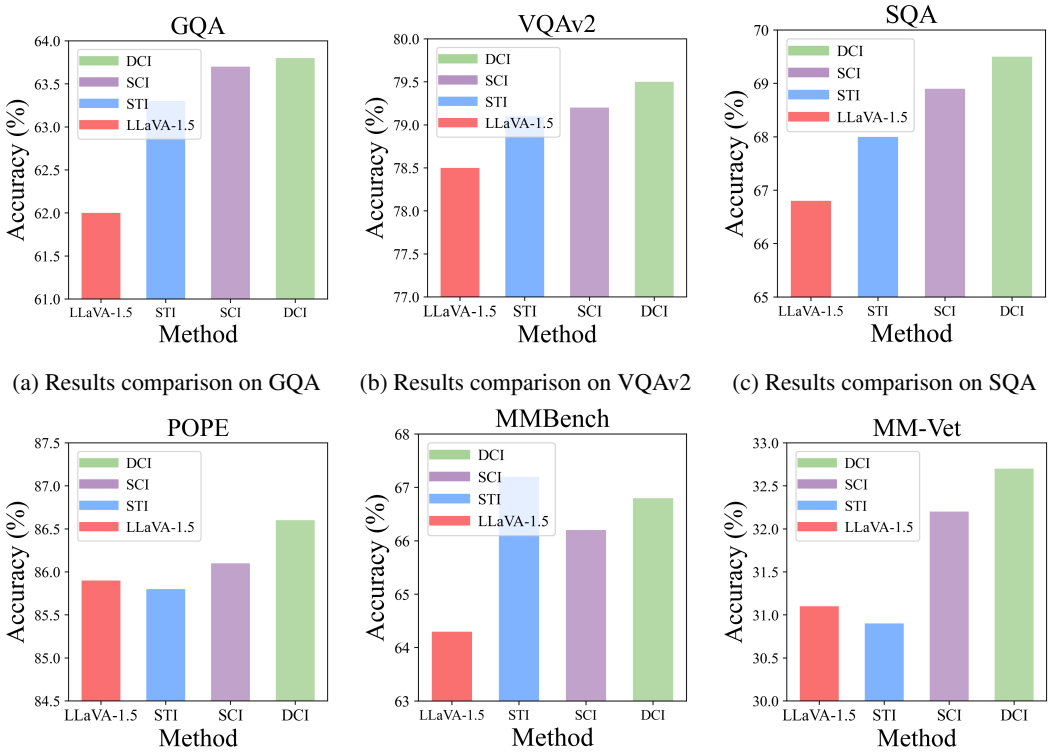
where $Ln(\cdot)$ and $Linear(\cdot)$ denotes Layer Normalization and Linear layer, respectively.

**Experimental Results and Analysis.** We present the detailed results of these described instantiations in Tab. 6. 1) For the STI method, using features from the 8th, 16th, and 24th layers, average pooling demonstrates superior performance compared to 1D convolutional downsampling, especially on the GQA [58] and MMBench [64] benchmarks. 2) In the SCI method, applying 2D convolution to enhance local feature modeling with offline ViT features from the 8th, 16th, and 24th layers achieves comparable performance to methods without 2D convolution. 3) Furthermore, incorporating additional aggregation information, such as a linear layer, into the DCI method does not lead to improved outcomes.

In summary, our attempts to introduce additional parameterized modules did not yield the anticipated improvements. We hypothesize that since the connector is randomly initialized, maintaining ease of convergence is crucial for effectively training the connector to align visual and language models under

Table 6: Additional studies on the Dense Connector.

| Architecture | Layer Index | GQA | VQA$^{v2}$ | SQA$^{I}$ | MMB | MMVet |
|---|---|---|---|---|---|---|
| LLaVA [16] | 24 | 62.0 | 78.5 | 66.8 | 64.3 | 31.1 |
| + *STI* | 8,16,20,24 | 63.0 | 79.1 | 68.0 | **67.6** | 30.8 |
| + *STI* | 8,16,24 | **63.3** | **79.1** | **68.0** | 67.2 | **30.9** |
| + *STI w/* 1D Conv | 8,16,24 | 62.6 | 78.9 | 67.4 | 65.0 | 30.6 |
| + *SCI* | 8,16,24 | **63.7** | **79.2** | **68.9** | **66.2** | **32.2** |
| + *SCI* | 4,8,16,24 | 62.9 | 78.9 | 68.3 | 65.2 | 30.2 |
| + *SCI* | 8,12,16,24 | 63.5 | 79.0 | 67.4 | 65.6 | 31.7 |
| + *SCI* | 8,16,20,24 | 63.6 | 79.2 | 67.0 | 65.8 | 31.9 |
| + *SCI w/* 2D Conv | 8,16,24 | 63.6 | 79.0 | 68.1 | 66.0 | 32.2 |
| + *DCI* | (1-12),(13-24) | **63.8** | **79.5** | **69.5** | **66.8** | 32.7 |
| + *DCI w/* Linear | (1-12),(13-24) | 63.7 | 79.4 | 68.5 | 66.4 | **32.8** |




(a) Results comparison on GQA (b) Results comparison on VQAv2 (c) Results comparison on SQA

(d) Results comparison on POPE (e) Results comparison on MMB (f) Results comparison on MM-Vet

Figure 4: Comparison of three instantiations of Dense Connector with LLaVA-1.5. STI stands for Sparse Token Integration, SCI for Sparse Channel Integration and DCI for Dense Channel Integration.

limited training data. In current MLLMs, the MLP structure of LLaVA [16] efficiently facilitates convergence between visual and language components. However, adding extra parameters may disrupt this inherent characteristic, leading to diminished performance. Future research will explore more complex and effective implementations of the Dense Connector.

### A.2 Visualized Analysis

In Appendix A.1, we further explored more complex Dense Connector structures. However, we found that non-parameterized methods yielded better average results. Therefore, we also visualized the comparison of three main instantiations of the Dense Connector with LLaVA-1.5 in Fig. 4, clearly demonstrating DCI's superior performance across different benchmarks.


https://doi.org/10.52202/079017-1043

Table 7: More comprehensive evaluation results of the Dense Connector. "PT", "IT", and "Res." denote pre-training data, instruction fine-tuning data, and image resolution, respectively. ‡ indicates the pre-training data (0.5M) from LLaVA-1.5 [16] and instruction data (1.5M) from Mini-Gemini [18].

| Method | PT+IT | Res. | LLM | GQA | VQA$^T$ | SQA$^I$ | MMB | MME$^p$ | MMMU$^v$ | Math | MMV | LLaVA$^W$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dense Connector | 0.5M+0.6M | 384 | Phi2-2.7B | 61.5 | 55.8 | 70.3 | 70.5 | 1487 | 36.6 | 28.2 | 33.8 | 65.1 |
| Dense Connector | 0.5M+0.6M | 384 | Vicuna-7B | 64.4 | 62.6 | 70.5 | 68.4 | 1523 | 36.7 | 25.5 | 35.4 | 67.4 |
| Dense Connector | 0.5M+0.6M | 384 | Llama3-8B | 65.1 | 62.2 | 75.2 | 74.4 | 1558 | 40.4 | 28.6 | 34.6 | 68.8 |
| Dense Connector | 0.5M+0.6M | 384 | Vicuna-13B | **65.4** | 64.7 | 73.0 | 71.4 | 1569 | 34.3 | 29.6 | 41.6 | 73.6 |
| Dense Connector | 1.2M+1.5M | 384 | Vicuna-13B | 64.6 | 65.0 | 77.1 | 74.4 | 1579 | 37.2 | 36.5 | 47.8 | 88.9 |
| Dense Connector | 0.5M+0.6M | 384 | Yi-34B$_{LoRA}$ | 63.9 | 66.7 | 80.5 | 77.7 | 1588 | **47.1** | 33.5 | 41.0 | 75.1 |
| Dense Connector | 0.5M+0.6M | 384 | Llama3-70B$_{LoRA}$ | 64.0 | 66.0 | **82.4** | 79.4 | 1622 | 47.0 | 32.9 | 46.1 | 74.5 |
| Dense Connector | 0.5M+1.5M‡ | 384 | Llama3-70B$_{LoRA}$ | 64.1 | **68.3** | 74.5 | **80.2** | **1649** | 43.6 | **40.7** | **53.3** | **97.8** |

Table 8: Ablation study on fine-tuning Vision Transformer. In this table, all the results are conducted using LLaVA 1.5 data, comprising 558K pre-training data and 665K instruction-tuning data.

| Method | VE | FT-ViT | Res. | LLM | **GQA** | **SQA$^I$** | **VQA$^T$** | **MMB** | **MMV** | **MMMU$^v$** | **Math** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DC (*w/* LLaVA) | CLIP-L | ✗ | 336 | Vicuna-7B | 63.8 | 69.5 | 59.2 | 66.8 | 32.7 | 34.8 | 26.9 |
| DC (*w/* LLaVA) | CLIP-L | ✓ | 336 | Vicuna-7B | 63.7 | 67.4 | 60.2 | 68.6 | 34.4 | 35.4 | 26.0 |
| DC (*w/* LLaVA) | Siglip-SO | ✗ | 384 | Vicuna-7B | 64.2 | 70.5 | 62.6 | 68.4 | 35.4 | 36.7 | 25.5 |
| DC (*w/* LLaVA) | Siglip-SO | ✓ | 384 | Vicuna-7B | 65.0 | 71.6 | 63.4 | 69.3 | 35.8 | 35.2 | 27.0 |

### A.3 Model Zoo

To probe the upper limits of MLLMs, we attempt to utilize a larger dataset to fine-tune our model based on Llama3-70B-Instruct. Given the limitations of computational resources, we commence with the Dense Connector pre-trained on LLaVA's initial stage data [16] and subsequently refine it using the instructional dataset from Mini-Gemini [18]. This model achieves accuracy of 40.7% on the MathVista [63] and 53.3% on MM-Vet [65], as illustrated in Tab. 7. Remarkably, our 70B model exhibits performance that is on par with GPT-4V on the LLaVA-Bench-in-the-Wild evaluation [16]. Specifically, our model achieves 97.8%, closely trailing GPT-4V's 98% [85]. However, it is worth noting that the full potential of our 70B model remains untapped due to the lack of initial stage pre-training data and the inherent constraints of LoRA fine-tuning [86].

### A.4 Further Exploration in Training Visual Encoder

In the experiments above, we used frozen multi-layer features from a pre-trained ViT to enhance the model's visual perception capabilities. Recently, however, there has been a growing trend of fine-tuning the entire model, including ViT, during the instruction-tuning phase to achieve better results. Following this trend, we present additional results in Tab. 8 on the performance impact of fine-tuning the ViT within the multi-layer visual connector. Specifically, in the first stage, we maintain the original training strategy. In the second stage, we fine-tune the ViT with a smaller learning rate of 2e-6, keeping other hyperparameters unchanged. Tab. 8 shows that fine-tuning ViT yields performance gains on benchmarks with a higher reliance on visual information, such as MMBench [64] and TextVQA [64]. For instance, fine-tuning CLIP-Large encoder resulted in improvements of 1.8% and 1.0% on MMBench and TextVQA, respectively. Conversely, on benchmarks that are less visually dependent, such as SQA [60], performance exhibited a slight decline.

### A.5 More Qualitative Results

From Fig. 6 to Fig. 13, we present additional multimodal dialogue illustrations with images and videos. Fine-tuned on image-text datasets, Dense Connector achieves effective alignment between visual and linguistic modalities, enhancing the model's visual perception capabilities. Qualitative evaluations showcases that our model can adeptly handle a broad spectrum of challenges across various domains, including mathematics, flowcharts, and daily scenarios. Additionally, the incorporation of the FreeVA method [54] equips our model with video understanding capabilities, enabling our model to address dynamic temporal challenges. However, due to the lack of training on video data, our model may occasionally misinterpret images as videos. For instance, as shown in Fig. 12, the model mistakenly interprets videos as images, incorrectly responding with 'The image' instead of 'The video'.

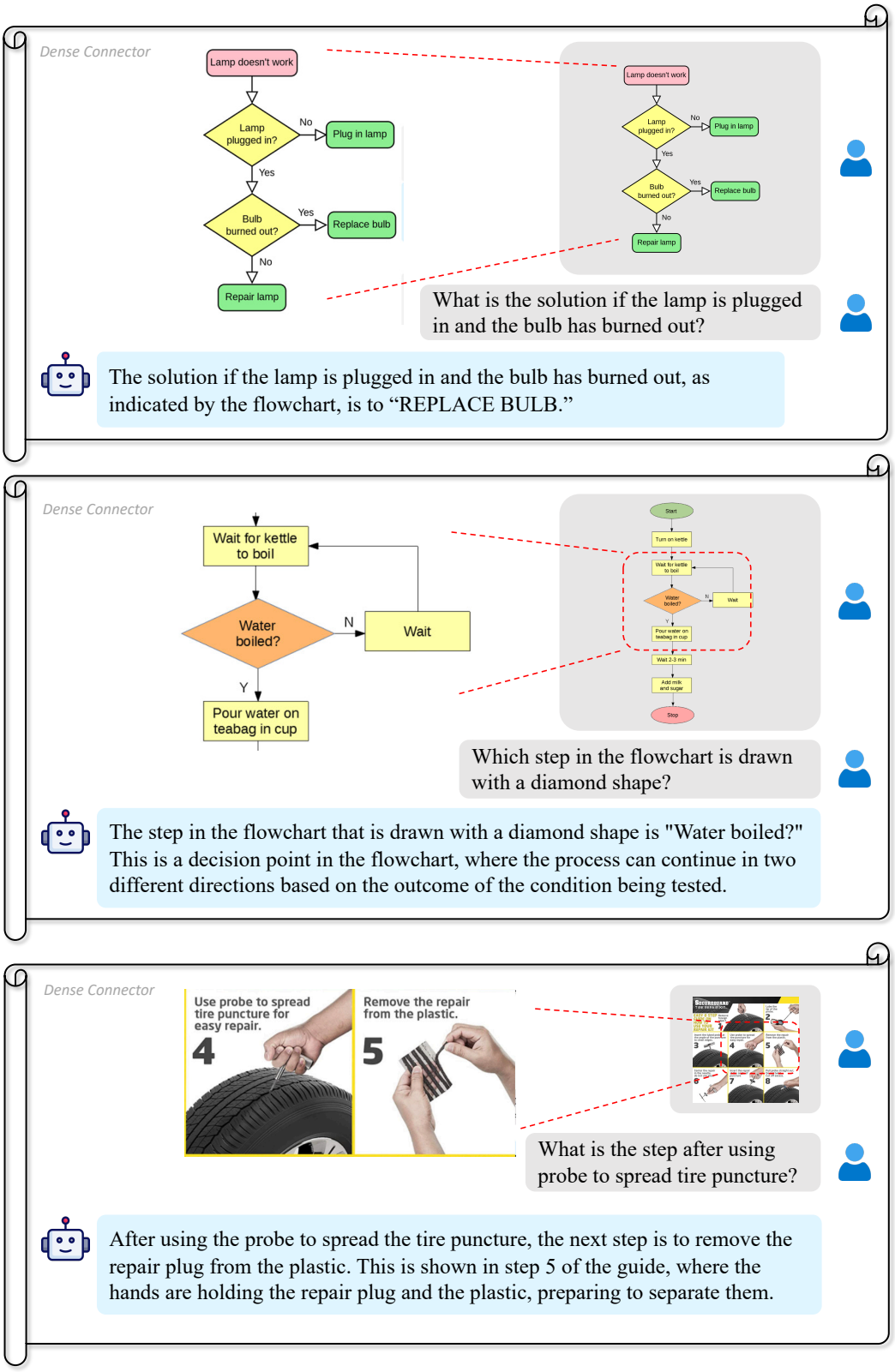



Figure 5: Qualitative results of the flowchart understanding.


https://doi.org/10.52202/079017-1043

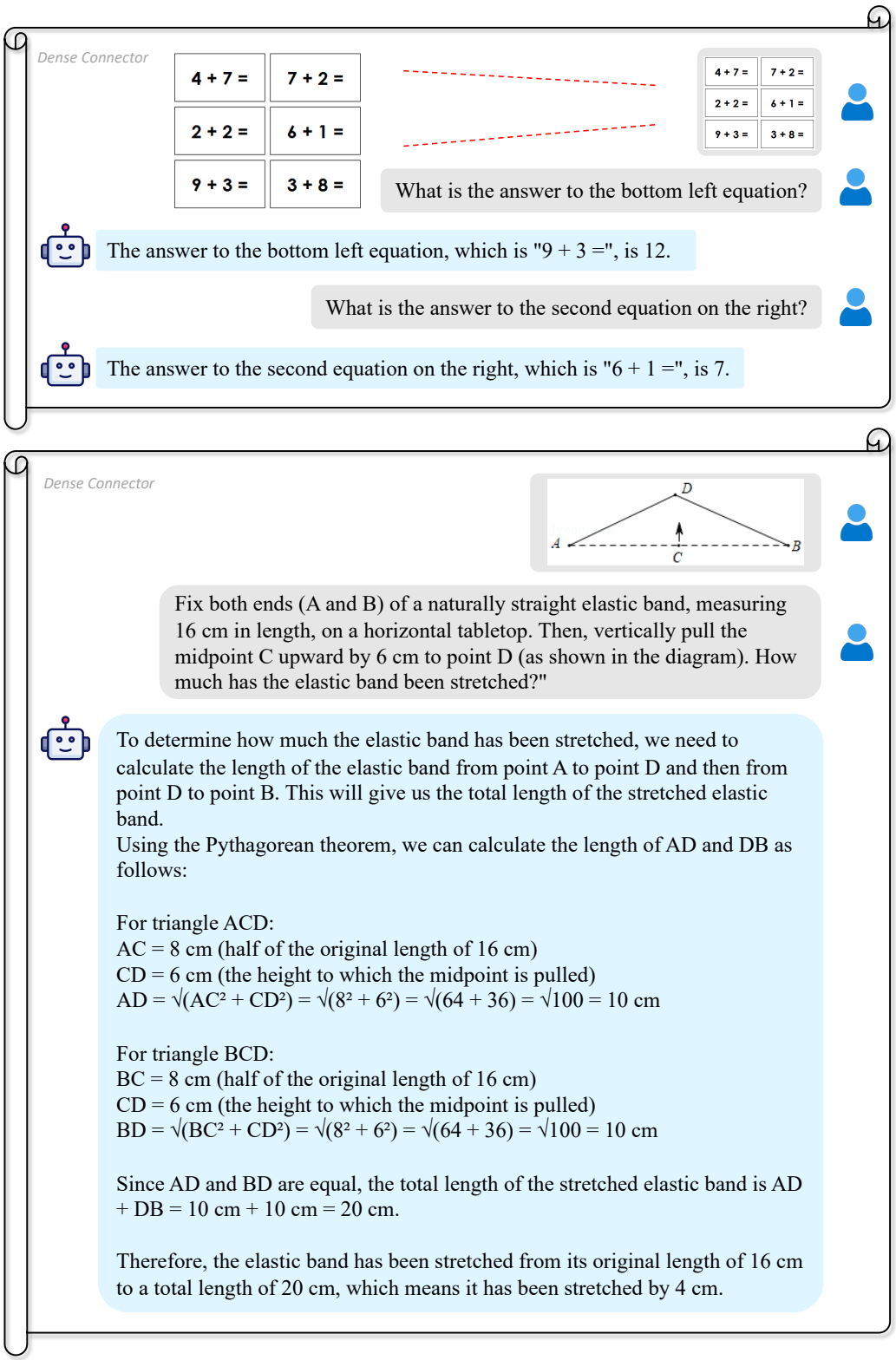



Figure 6: Qualitative results of the math solving skills.

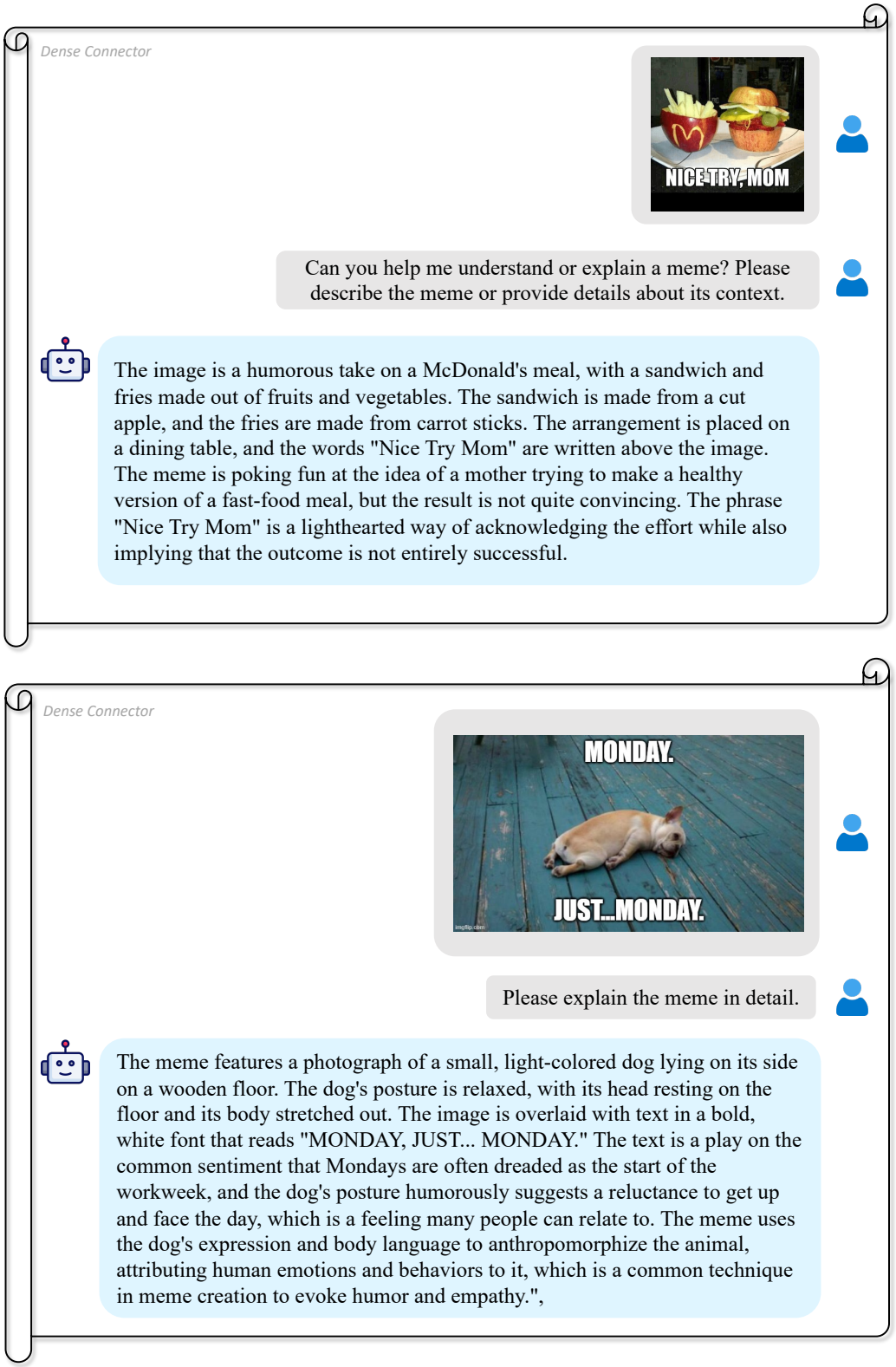



Figure 7: Qualitative results of the meme understanding.


https://doi.org/10.52202/079017-1043

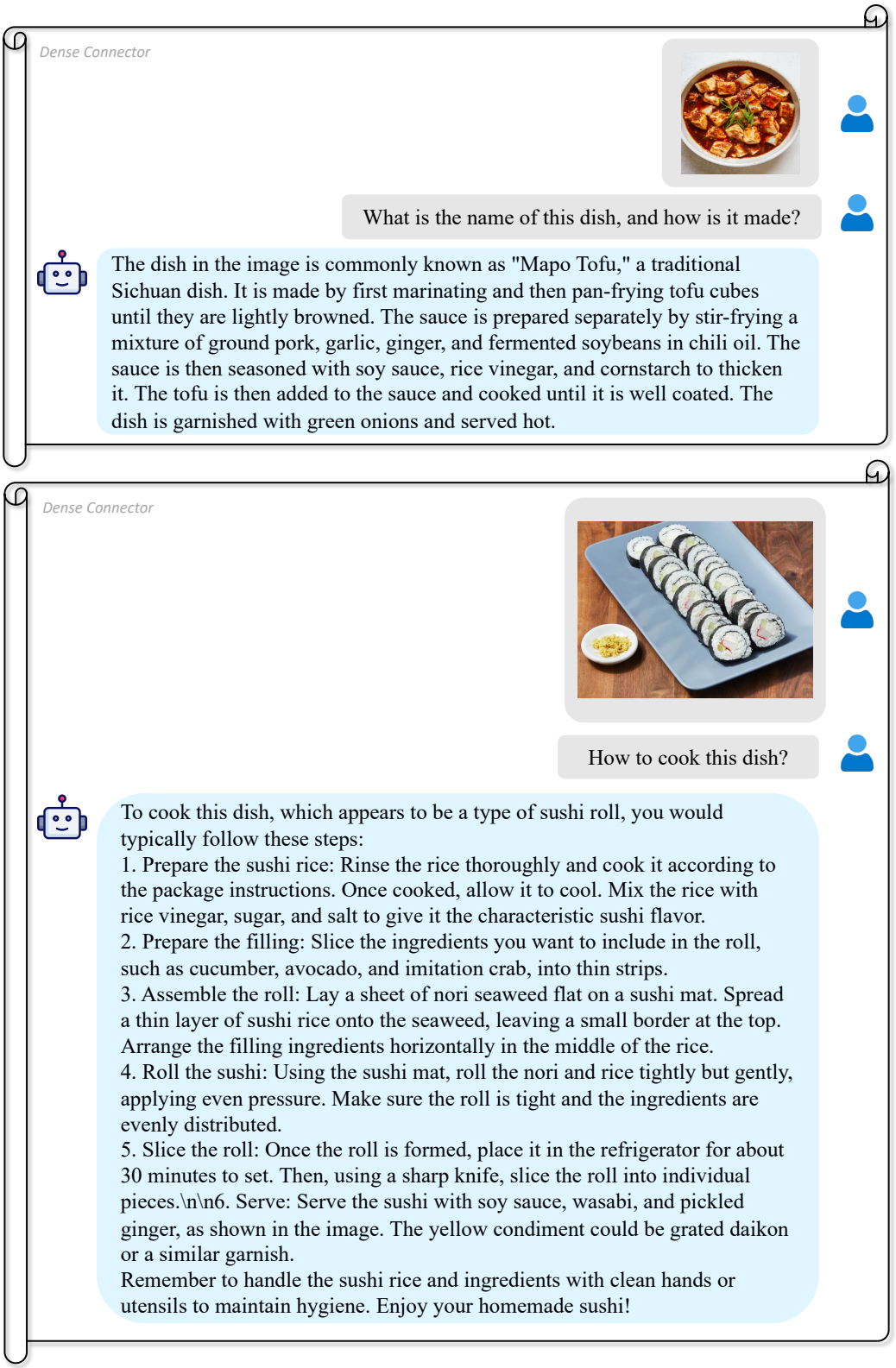



Figure 8: Qualitative results of the delicacies understanding.

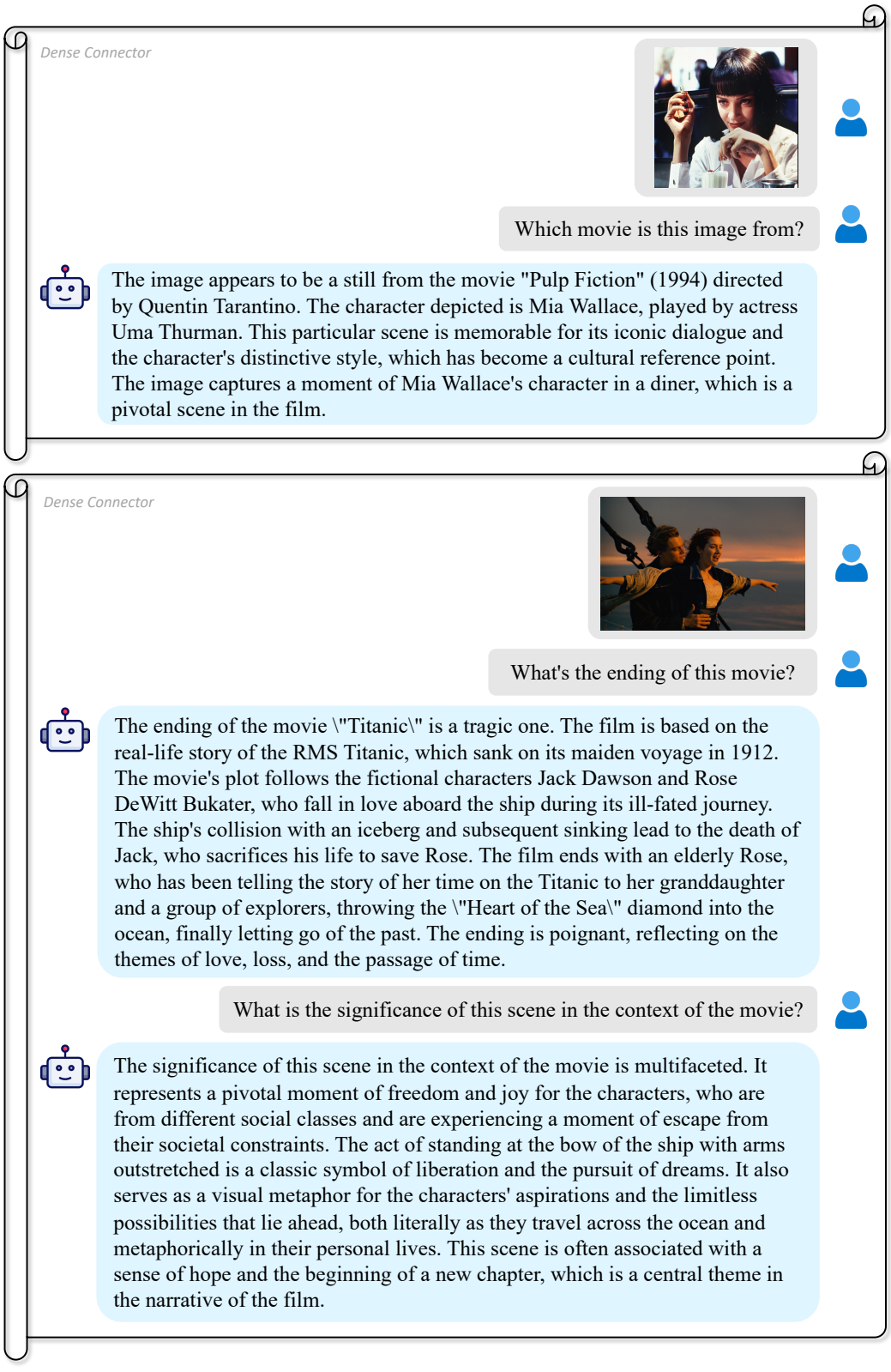



Figure 9: Qualitative results of the movie analysis.


https://doi.org/10.52202/079017-1043

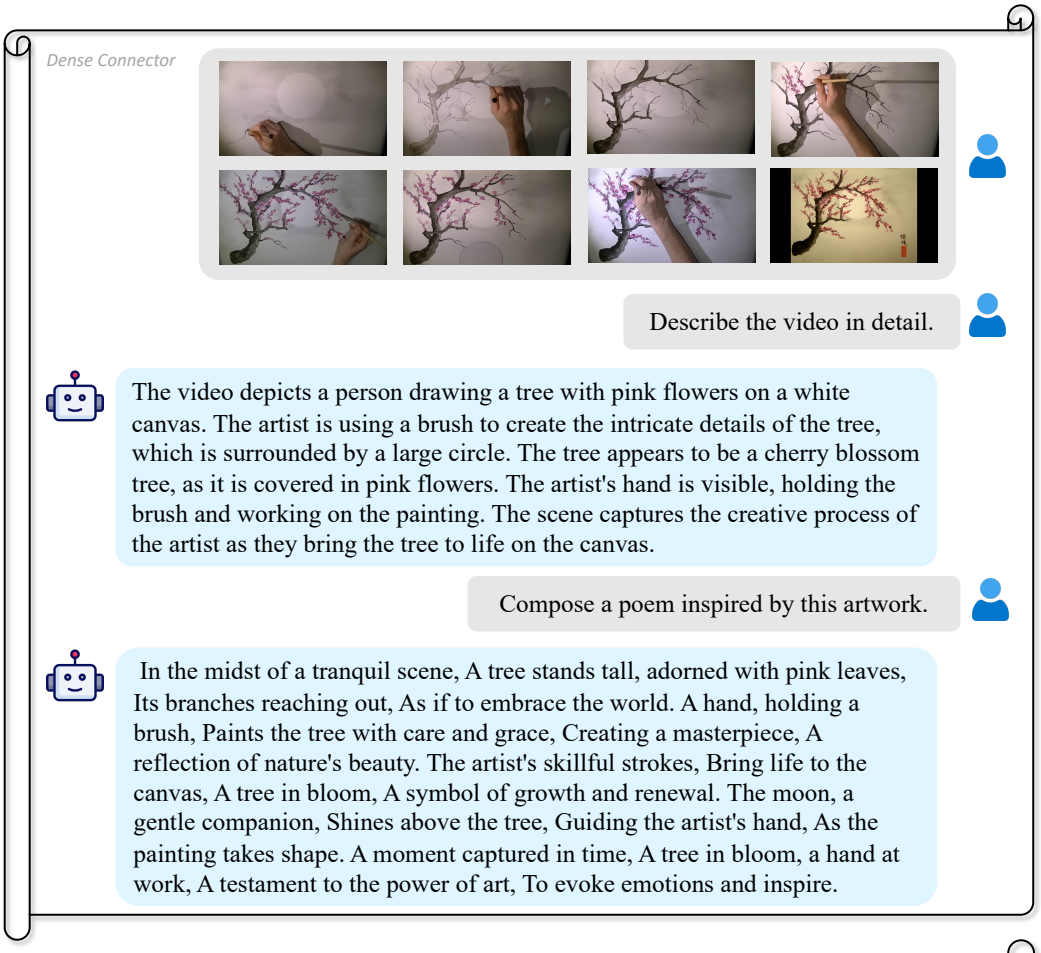



Figure 10: Qualitative results of the video understanding.

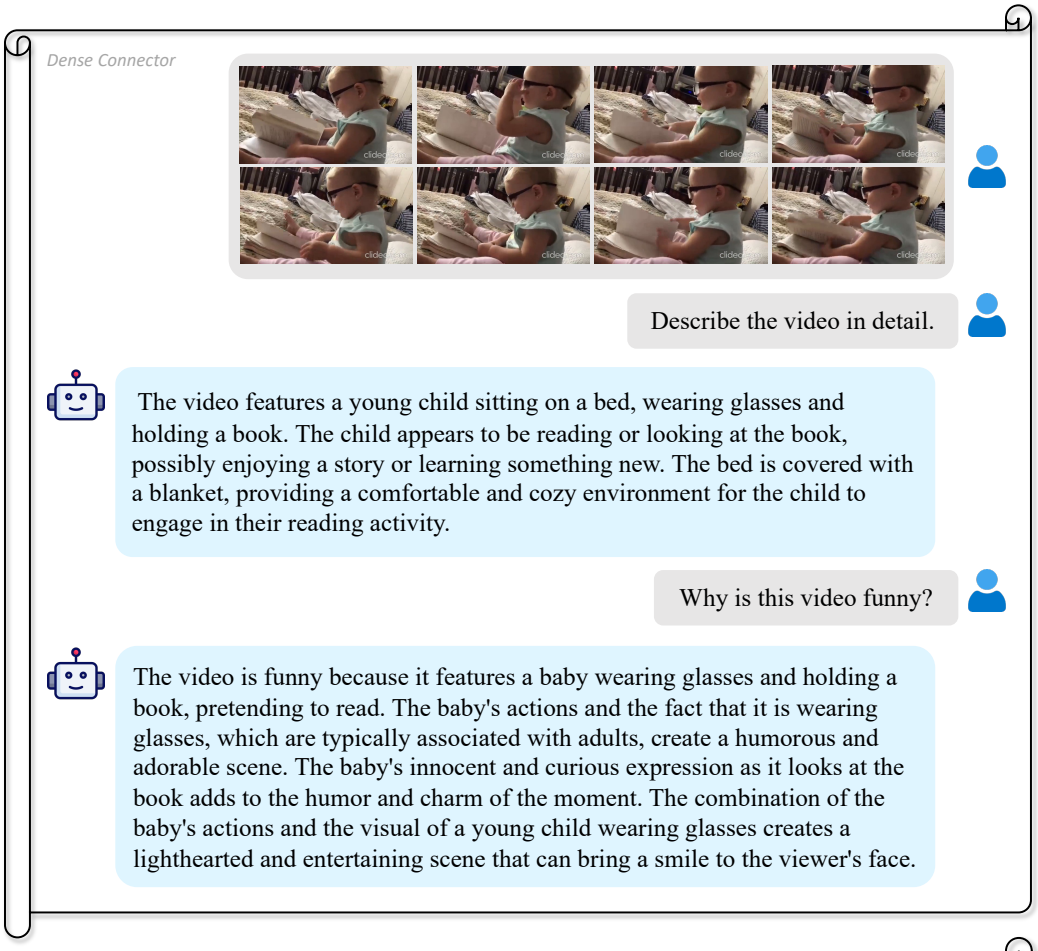



Figure 11: Qualitative results of the video understanding.


https://doi.org/10.52202/079017-1043

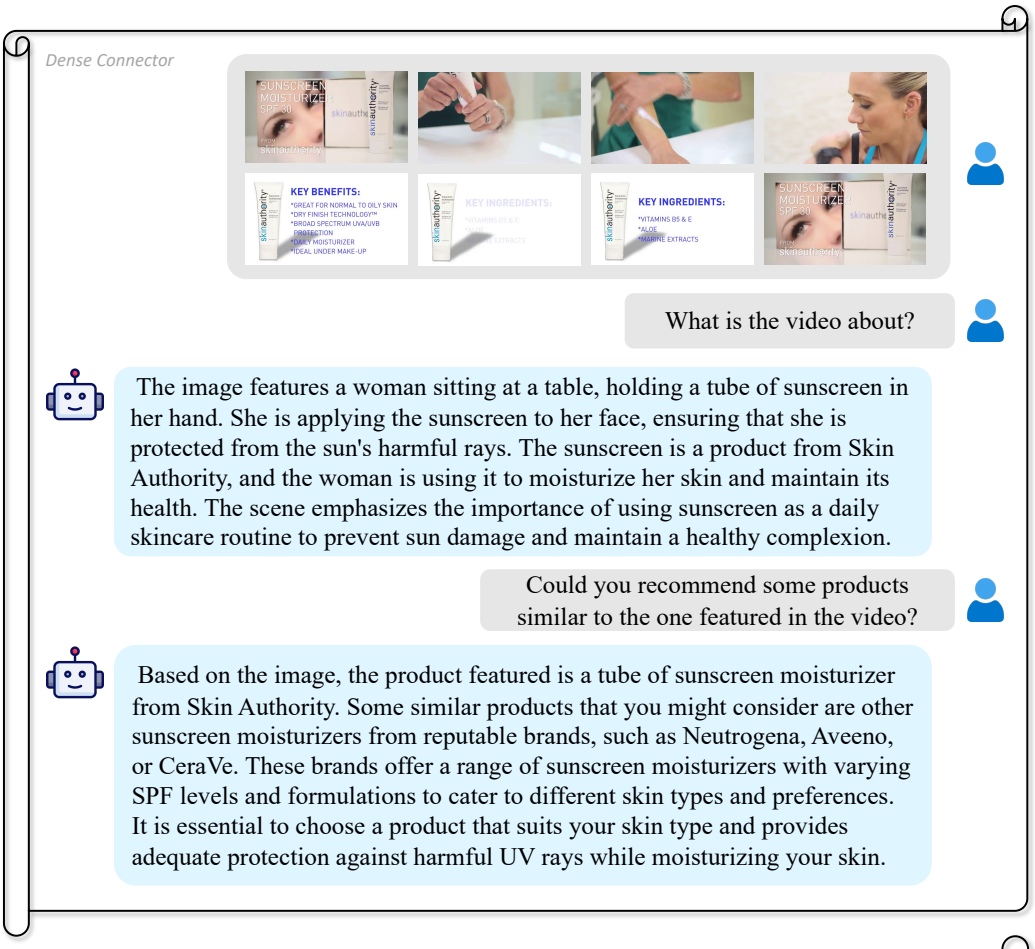



Figure 12: Qualitative results of the video understanding.

https://doi.org/10.52202/079017-1043

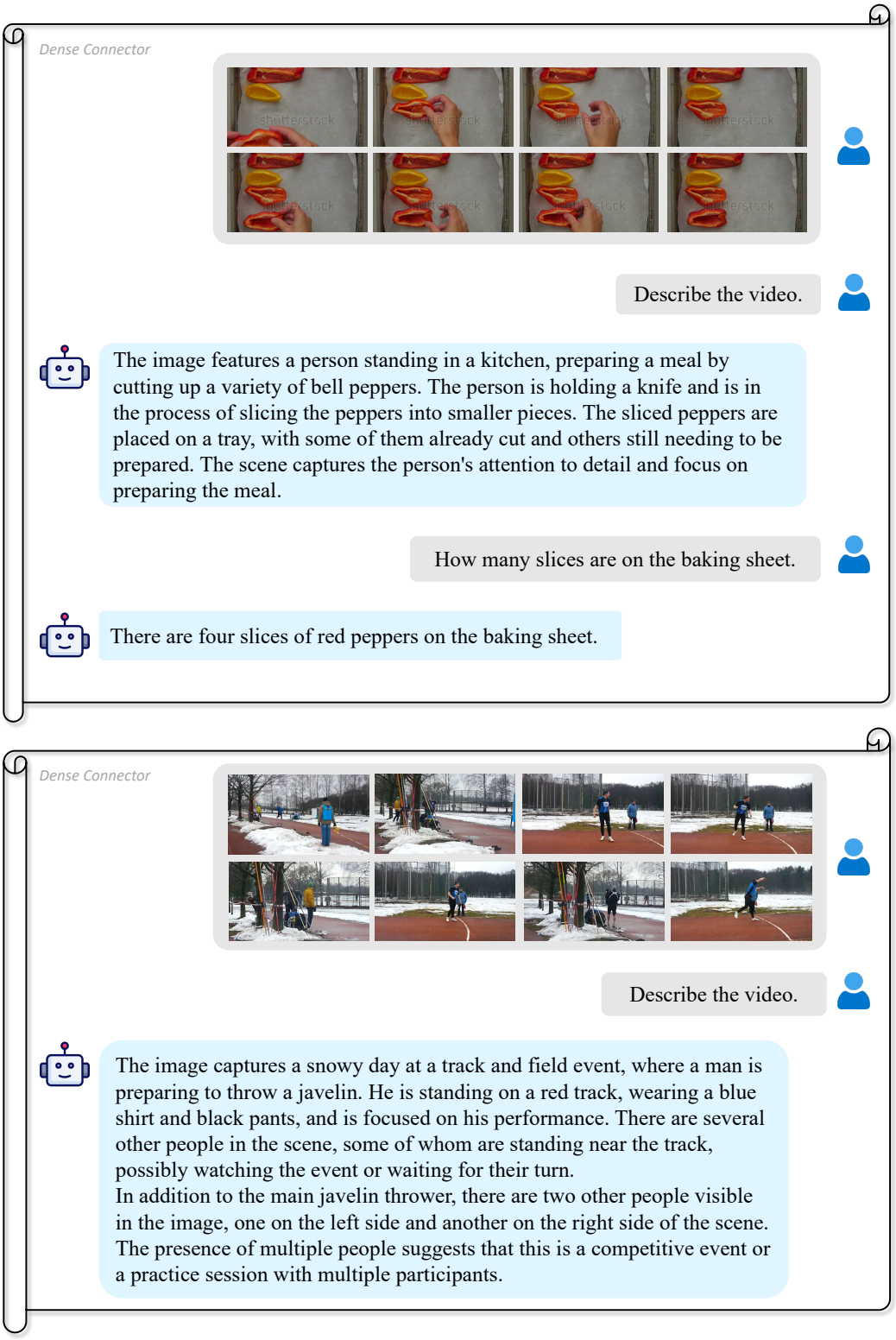



Figure 13: Qualitative results of the video understanding.


https://doi.org/10.52202/079017-1043

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We provide our contributions both in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitation in the paper.

   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: Our paper does not include theoretical 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: Yes, we provide all necessary information to reproduce the experimental results.

   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?


https://doi.org/10.52202/079017-1043

Answer: [Yes]

Justification: We will provide the anonymous code, and our code and data will be publicly available.

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: The training and test details are described in the paper.

   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 because it would be too computationally expensive.

   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: We provide these information in the implementation details.

   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: Our research follows 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: There is no societal impact of the work performed.

    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.


https://doi.org/10.52202/079017-1043

- 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: The paper poses no such 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 original paper for assets.

    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: [Yes]

    Justification: We will provide an anonymous URL.

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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


https://doi.org/10.52202/079017-1043