We are genuinely grateful for the reviewer's thorough and insightful comments, which have contributed significantly to improving the quality of our paper. In our efforts to share our insights and the notably effective design with the research community, we have significantly enhanced the overall quality of the manuscript over the past month. Additionally, we are committed to addressing the concerns raised by the reviewer in the upcoming revised version.

W1:

(a) We agree with the reviewer that several other factors affect the performance of MLLMs, such as the capability of the language model, the temperature parameter used during generation, and so forth. However, our primary focus is on examining the critical factors involved in augmenting LLMs with visual perception and cognition abilities to develop MLLMs. Specifically, we believe that (1) the alignment of vision-language representations and (2) the alignment of the LLM's comprehension with visual semantics encompass the core process of establishing the visual perception and cognitive skills of MLLMs. Therefore, we assume that other influencing factors, apart from the visual component, remain constant. We will include this clarification in the revised manuscript in hopes of addressing the reviewer's concerns.

(b) We apologize for any confusion caused by the phrasing in the current version. Our intended meaning is as follows: Images contain rich visual semantic information, including but not limited to object attributes, spatial relationships between objects, sizes, and colors. Capturing all of these visual semantics through textual descriptions is challenging, and doing so may lead to the loss of valuable visual information. Therefore, the primary aim of vision-language alignment should focus on distributional alignment, which enhances the processing of visual representations by LLMs, rather than on semantic alignment, which attempts to translate visual semantics into textual semantics. Additionally, many existing approaches utilize relatively simple projection layers (such as MLPs or linear layers) for achieving vision-language alignment, which, given their capabilities, are better suited for distributional alignment than semantic alignment.

In response to the reviewer's suggestions, we have included two sets of visual analyses. The first set (ex1) presents a comparison of visual token distributions before and after projection with text token distributions, while the second set (ex2) examines the semantic similarity between the projected visual tokens and both the visual tokens before projection and the text tokens. Specifically, we randomly selected 100 images and generated corresponding textual captions using a well-trained LLaVA-Next model. For experiment (ex1), we utilized PCA to reduce all token embeddings to 2D and visualize their distributions. In experiment (ex2), to evaluate the semantic similarity between a projected visual token and its counterpart before projection, we defined the attribute vector of a visual token as its cosine similarity to all visual tokens derived from the same image. Subsequently, the semantic similarity between visual tokens is calculated by computing the cosine similarity of their attribute vectors. Here is the formal mathematical definition:

1. *Defining the Attribute Vector*:
   Suppose we have a visual token v_i from image I, and there are N visual tokens in total from image I, denoted as $\{v_1, v_2, \ldots, v_N\}$. Then the attribute vector $\mathbf{a}_i$ of v_i can be represented as:

   $$\mathbf{a}_i = [\cos(\theta_{i1}), \cos(\theta_{i2}), \ldots, \cos(\theta_{iN})]$$

   Where $\cos(\theta_{ij})$ is the cosine similarity between visual tokens v_i and v_j, calculated as:

   $$\cos(\theta_{ij}) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|}$$

2. *Calculating Semantic Similarity between Visual Tokens*:
   The semantic similarity between two visual tokens v_i and v_j is obtained by computing the cosine similarity of their attribute vectors. Suppose $\mathbf{a}_i$ and $\mathbf{a}_j$ are the attribute vectors of v_i and v_j respectively, then the semantic similarity $S(v_i, v_j)$ between them can be expressed as:

   $$S(v_i, v_j) = \frac{\mathbf{a}_i \cdot \mathbf{a}_j}{\|\mathbf{a}_i\| \|\mathbf{a}_j\|}$$

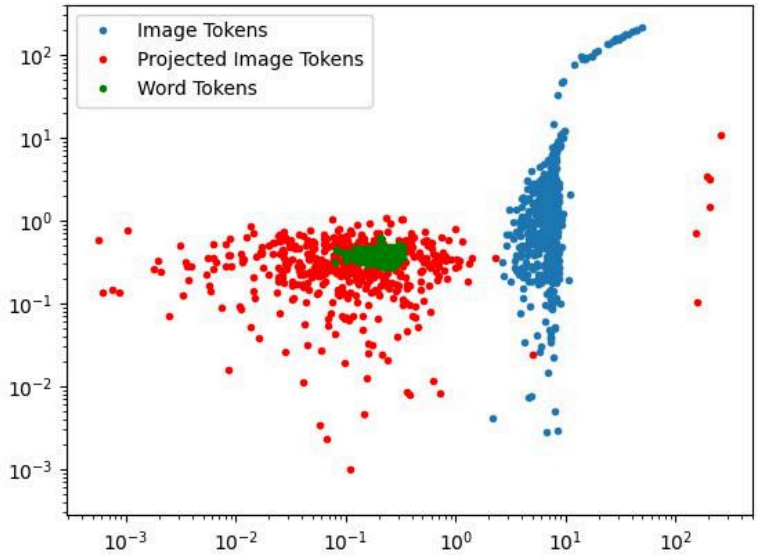
Furthermore, we define the semantic similarity between the projected visual tokens and textual tokens as the maximum cosine similarity between the visual token and all textual tokens in the caption of its source image.

3. *Calculating Semantic Similarity between Visual and Textual Tokens*:
   Suppose we have a visual token v_i from image I, and there I has a caption with N textual tokens $\{t_1, t_2, \ldots, t_n\}$. The semantic similarity $S(v_i, \{t_1, t_2, \ldots, t_n\})$ between the visual token v_i and textual tokens is:

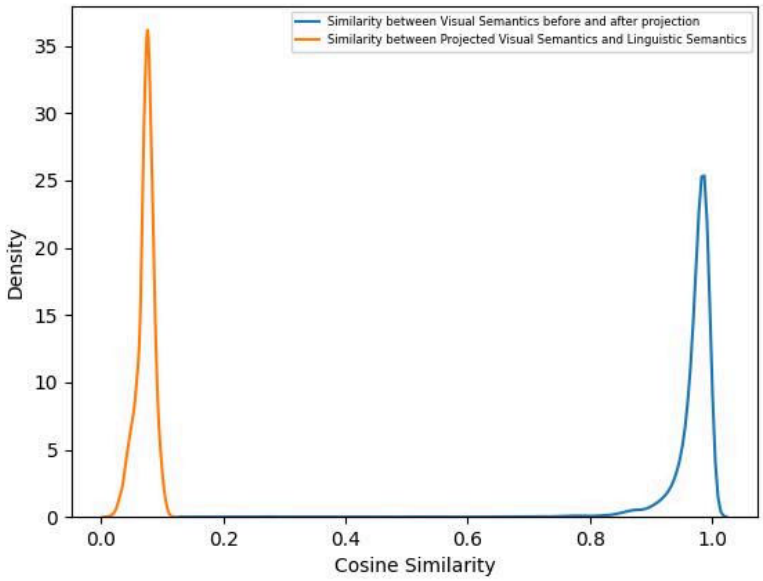
$$S(v_i, \{t_1, t_2, \ldots, t_n\}) = \max_{j \in \{1,2,\ldots,n\}} \left(\frac{\mathbf{v}_i \cdot \mathbf{t}_j}{\|\mathbf{v}_i\| \|\mathbf{t}_j\|} \right)$$

We report the visualized results of (ex1) and (ex2):

(Figure ex1) Comparison of visual token distributions before and after projection with text token distributions:



(Figure ex2) Comparison of semantic similarities between the projected visual tokens and both the visual tokens before projection and the text tokens.



Our findings from Experiment 1 indicate that, although not yet perfect, the current visual-language pre-alignment training effectively maps the distribution of visual tokens into a space similar to that of textual tokens. Furthermore, Experiment 2 demonstrates that the projected visual tokens maintain a high degree of semantic similarity with their counterparts before projection, but they do not establish a direct semantic correspondence with the textual tokens.

W2:

(a) Specifically, we design SA-Perceiver, which comprising four $\mathbb{R}^{1024\times1024}$ linear layers and one $\mathbb{R}^{1024\times4096}$ linear layer (save 60% parameters compared with the projection module in LLaVA-Next, which consists of a $\mathbb{R}^{1024\times4096}$ and a $\mathbb{R}^{4096\times4096}$ linear layer), to integrate high-resolution image information into low-resolution image features at a lower cost. Only the low-resolution features are then utilized as input to the LLM. Given that the projection module (including MLP, Q-former, and our SA-Perceiver) has a significantly lower parameter count and computational complexity

than the LLM, the overall system latency is predominantly dictated by the computation delay of the LLM. As SA-Perceiver enables a reduction of visual sequence length up to four-fold, and the time complexity of LLM is $O\left(n^2\right)$, our method can theoretically achieve a maximum reduction in latency by a factor of 16. However, system latency is also affected by factors such as the number of input text tokens, the length of the generated sentences, and other intricate system dynamics. To more accurately assess the impact of SA-Perceiver in reducing computational overhead, we randomly select 1,000 images, remove their textual instructions, resize them to various resolutions, and compare the latency and FLOPs of our method and LLaVA-Next during the feedforward process.

Latency of processing 1000 images (seconds):

Method	**336x336**	**672x336**	**1008x336**	**672x672**
LLaVA-Next	449	475	564	738
VLSA	**373**	**377**	**385**	**399**

FLOPs in processing 1000 images (GFLOPs):

Method	**336x336**	**672x336**	**1008x336**	**672x672**
LLaVA-Next	18798.9	27444.3	36462.3	45840.6
VLSA	**9842.3**	**9884.7**	**10190.8**	**10539.3**

(b) To comprehensively address the concerns about the architecture of SA-Perciever, we first supplement the justification and ablations about the current design and then report the performances on additional ablations provided by the reviewer.

The SA-Perceiver is designed to integrate features from high-resolution images into the features of low-resolution images. To achieve this, we first implement a cross-attention layer that collects information from high-resolution images. As these high-resolution images are divided into multiple sub-images during preprocessing, we have subsequently incorporated a self-attention layer to enhance the modeling of interrelationships among the sub-image information. To ensure parameter efficiency, we have omitted certain projection layers typically found in the standard attention mechanism. To demonstrate the effectiveness of these design choices, we conducted the following ablation experiments on SA-Perceiver: (ex3) removing the self-attention layer and (ex4) retaining all linear projections in cross and self-attention (including those for key, query, value, and output).

Variant	**GQA**	**SQA-I**	**DocVQA**
(ex3) w/o. self-attn	65.1	76.9	72.4
(ex4) full projections	65.0	77.3	**75.3**
VLSA	**65.3**	**77.5**	75.2

Our rationale for introducing the learnable parameter P stems from the requirement of the text-to-image model, stable diffusion 3-medium, utilized in our reconstruction training, which necessitates a pooled embedding as input to encapsulate global semantic information (this was implicitly mentioned in line 250, but we will clarify it further). To better substantiate the efficacy of learnable parameter P, we have included an ablation study employing global pooling on the aligned visual tokens to generate the pooled embedding.

Variant	**GQA**	**SQA-I**	**DocVQA**
(ex5) Global Pooling	64.9	76.2	73.3
Learnable Parameter	**65.3**	**77.5**	**75.2**

In response to the reviewer's suggestion regarding the three ablation studies, we employed global average pooling to compute the pooled embedding and subsequently report on their performance.

Variant	**GQA**	**SQA-I**	**DocVQA**
(ex6) High-Res. Only	62.1	70.1	73.9
(ex7) High-Res. + Self-Attn	62.3	69.6	74.1
(ex8) [High-Res., Low-Res.] + Self-Attn	**65.5**	76.9	69.8
LLaVA-Next *([High-Res., Low-Res.])*	64.6	75.1	73.7
VLSA	65.3	**77.5**	**75.2**

Before delving into the analysis of these ablation results, it is essential to highlight that none of the variants effectively address the surge in computational costs associated with high-resolution inputs, which serves as a key motivation for our sa-perceiver. By comparing (ex6), (ex7), and other findings, we observe that the lack of low-resolution inputs adversely impacts the performance of MLLM on general VQA tasks. Alongside the experimental results presented in the main text, we hypothesize that low-resolution inputs enhance MLLM's capacity to perceive global semantics, while high-resolution inputs augment its ability to discern fine-grained semantics. Notably, we find that employing self-attention in (ex8) to enhance the interaction between high and low-resolution features improves global semantic perception, yet significantly hinders performance on tasks necessitating fine-grained perception, such as document understanding. In contrast, our VLSA approach provides a balanced enhancement of various capabilities while conserving computational resources.

(C) Following the constructive suggestions, we have conducted ablations on the two tasks within cognition alignment and provided some observational conclusions.

Ablation study on the two tasks of fine-tuning: (ex9) predicting Codebook indices and (ex10) predicting RGB pixel values.

Variant	Predict RGB	Predict Codebook indices	AI2D	SQA-I	ChartQA
w/o. Cognition Alignment	✕	✕	68.2	74.1	67.4
(ex9) w/o. RGB	✕	✓	69.8	77.2	67.7
(ex10) w/o. Codebook	✓	✕	68.4	71.6	67.5
VLSA	✓	✓	**71.4**	**77.5**	**67.9**

By comparing the results without cognition alignment to (ex9) and (ex10), we observe that predicting RGB values alone enhances the model's fine-grained cognitive capabilities, benefiting document understanding tasks. However, it may overly emphasize low-level semantics, detrimentally affecting the understanding of high-level semantics and leading to a significant performance drop on SQA. In contrast, predicting codebook indices alone consistently improves performance across various tasks. This might be attributed to VQ-VAE, as an autoencoder, being able to balance semantics at different levels. Furthermore, combining both tasks yields further improvements.

(d)
We report the performance of VLSA with the replacement of the backbone model to LLaMA2-7B, Vicuna1.5-7B, and Vicuna1.5-13B. Additionally, we report the performance of LLaVA-Next with these backbones as references.

Variant	LLM	GQA	AI2D	DocVQA
LLaVA-Next	LLaMA2-7B	62.1	66.7	71.8
(ex11) VLSA	LLaMA2-7B	63.0	68.4	73.2
LLaVA-Next	Vicuna1.5-7B	62.2	66.4	72.5
(ex12) VLSA	Vicuna1.5-7B	63.6	67.5	74.6
LLaVA-Next	Vicuna1.5-13B	65.4	67.0	72.7
(ex13) VLSA	Vicuna1.5-13B	**67.2**	**69.2**	**76.8**

We also apply VLSA to Qwen-VL[1] and LLaMA-Adapter V2[2] to demonstrate generality, and reported the preliminary experimental results. More results and implementation details will be included in the revised manuscript.

Variant	GQA	ChartQA	DocVQA	SEED-Bench	MME	COCO Cap
Qwen-VL	59.3	65.7	65.1	56.3	-	-
(ex14) Qwen-VL + VLSA	**62.1**	**66.4**	**65.4**	**62.0**	-	-
LLaMA-Adapter V2	-	-	-	32.7	1221	122.2
(ex15) LLaMA-Adapter V2 + VLSA	-	-	-	41.3	**1475**	**143.1**

[1] Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond
[2] LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model

W3:

To address the reviewer's concern, we would like to clarify that incorporating additional pre-trained models did not significantly increase the training costs. Specifically:

(1) The VQ-VAE model is employed solely for re-annotating the open-source dataset, which involves collecting the necessary labels for cognitive alignment. It does not directly participate in the training process, thereby not introducing any additional computational overhead.

(2) When training the VLSA, the LDM is required to perform only a single denoising step per iteration, in contrast to the multi-step denoising process used for image generation. Consequently, its computational overhead is much lower than the feedforward process of the LLM, and the additional costs brought by LDM are substantially outweighed by the efficiencies gained through our compressive image encoding (SA-Perceiver). In reference to the comparison method outlined in W2(a), we have quantified the impact of incorporating reconstructive training on both latency and FLOPs. We also report the effects of reconstructive training on the total training time of instruction tuning stage (with our 980K dataset on 16 Nvidia A100). These results demonstrate that our method maintains both generality and scalability.

Latency of processing 1000 images (seconds):

Method	**336x336**	**672x336**	**1008x336**	**672x672**
LLaVA-Next	449	475	564	738
VLSA(w/o. Reconstruct)	**373**	**377**	**385**	**399**
VLSA(w/. Reconstruct)	391	394	408	426

FLOPs in processing 1000 images (GFLOPs):

Method	**336x336**	**672x336**	**1008x336**	**672x672**
LLaVA-Next	18798.9	27444.3	36462.3	45840.6
VLSA(w/o. Reconstruct)	**9842.3**	**9884.7**	**10190.8**	**10539.3**
VLSA(w/. Reconstruct)	11646.1	12118.5	12545.0	13443.8

Training time in instruction tuning stage (hours):

Method	**Training Time**
LLaVA-Next	27.3
VLSA(w/o. Reconstruct)	**14.2**
VLSA(w/. Reconstruct)	16.9

Furthermore, following the reviewer's suggestion, we conducted ablation experiments to evaluate the performance of VLSA using standard two-stage training (we drop our stage 2 training for VLSA) and a smaller LDM(using Stable Diffusion 1.5). The results demonstrate that our method still significantly improves performance. (The impact of removing reconstructive training or cognition alignment has already been included in Table 3.)

Variant	**GQA**	**SQA-I**	**DocVQA**
LLaVA-Next	64.6	75.1	73.7
(ex16) Two-Stage Training	65.2	77.0	74.6
(ex17) Smaller LDM	64.8	76.8	75.1
VLSA	**65.3**	**77.5**	**75.2**

W4:

Fixed!

Q1:

Our aim is to support MLLMs in achieving a comprehensive understanding of images while eliminating reliance on additional manual annotations. We initially explored using RGB values as labels, as this method provides a straightforward way for self-supervised semantic annotation. However, this approach does not consistently improve performance across various downstream tasks. We hypothesize that this limitation arises from RGB values only capturing superficial semantics, which may impede the grasp of deeper semantic concepts.

Consequently, we began investigating methods for annotating deep image semantics in a self-supervised manner. Then, pre-trained VQ-VAE emerged as an ideal solution. It provides discretized semantic labels that are compatible with textual input. Moreover, as an autoencoder, it effectively encodes and preserves original image information, aligning well with our objectives. Although VQ-VAE achieved good results, we found

that integrating these two approaches further improves performances. The underlying reason for this improvement may stem from relying exclusively on deep semantics such as codebook indices to comprehend shallow information is not straightforward, thereby necessitating a larger dataset for effective training. However, introducing the RGB value prediction task mitigates this issue.

Q2:

We had not previously attempted this approach, but it remains a highly intriguing endeavor. Our preliminary experiments indicate that this method enhances the convergence speed of the reconstruction loss during the initial training phase, although it does not seem to improve the final convergence performance. We believe that incorporating caption embeddings could alleviate the learning difficulties associated with epigone embeddings, as they would only need to capture the residuals between the caption embeddings and the original image semantics. However, acquiring image captions during inference still poses a significant challenge. (The authors intend to continue refining the implementation of this approach and will provide feedback to the reviewer if new results emerge during the rebuttal period.)