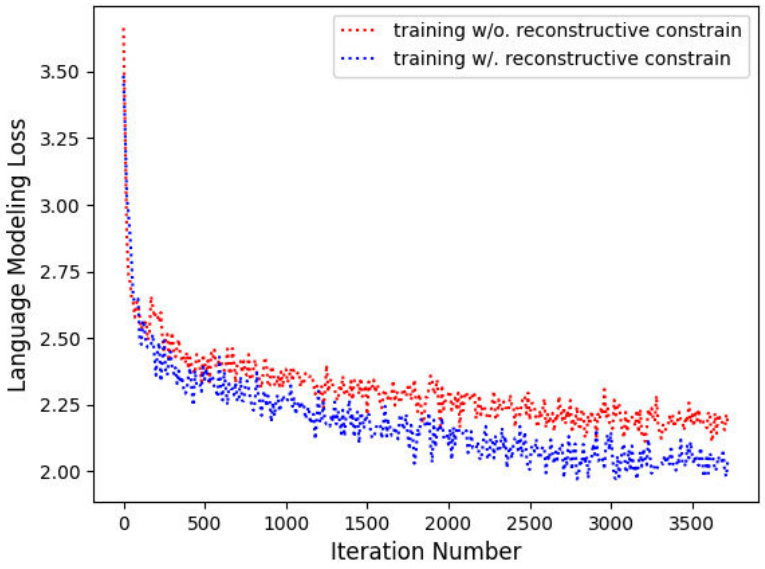
We sincerely appreciate the constructive feedback provided by the reviewer, which is invaluable for enhancing the quality of our paper. Below are our responses to the raised concerns. We also commit to accurately reflecting these points in the revised manuscript.

W1:

Our statement in the Limitation section may have been somewhat misleading. While it is indeed the case that the suboptimal combination strategy of the two types of loss can restrict VLSA from realizing its full potential, it is important to highlight that these two losses consistently operate synergistically, even when using the simplest balancing method, where the ratio between the losses is not adjusted, and both losses are applied throughout all training phases. To further illustrate this point, we have included Figure ex1, which depicts the language model loss curves before and after incorporating the reconstruction loss in Stage 1 training of VLSA. The results demonstrate that the introduction of the reconstruction loss leads to both a faster convergence and a more favorable final outcome for the language modeling loss. Therefore, our method does not hinder the optimization efficiency. Our primary focus in this article is to highlight the effectiveness of the proposed method, while further optimization will be addressed in future work.

(Figure ex1) The influence of reconstructive training on language modeling loss.



W2:

To address the reviewer's concern, we would like to **(a)** clarify that incorporating additional pre-trained models as well as two types of alignments did not significantly raise the training costs. (In fact, our method significantly improved the training efficiency compared to LLaVA-Next.) We will also demonstrate **(b)** the generality and scalability of VLSA by integrating it with various language models and different types of MLLMs, and adapting it to much higher input resolution.

(a):

(1) In cognition alignment, the VQ-VAE model is utilized exclusively for re-annotating the open-source dataset, which involves gathering the necessary labels for deep visual semantics without the involvement of human effort. It does not directly participate in the training process, thereby not introducing any additional computational overhead. Furthermore, cognition alignment is carried out concurrently with the standard instruction tuning process of MLLMs, without introducing any substantial new training phases. Therefore, cognition alignment has excellent scalability and is highly adaptable to other MLLM frameworks.

(2) In perception alignment, 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). We will first show that SA-Perceiver can significantly reduce computational overhead and enhance efficiency. Then, we will compare the computational efficiency of our VLSA (with LDM incorporated) with LLaVA-Next.

SA-Perceiver 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. In the same way, we have quantified the impact of incorporating reconstructive training on both latency and FLOPs. All these results are included in the following table.

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

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.

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

(b):

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
(ex2) VLSA	LLaMA2-7B	63.0	68.4	73.2
LLaVA-Next	Vicuna1.5-7B	62.2	66.4	72.5
(ex3) VLSA	Vicuna1.5-7B	63.6	67.5	74.6
LLaVA-Next	Vicuna1.5-13B	65.4	67.0	72.7
(ex4) 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	-	-
(ex5) Qwen-VL + VLSA	**62.1**	**66.4**	**65.4**	**62.0**	-	-
LLaMA-Adapter V2	-	-	-	32.7	1221	122.2
(ex6) 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

Finally, we increase the maximum input resolution of VLSA from 672x672 to 4096x4096, and report the preliminary experimental results.

Variant	Res.	GQA	AI2D	DocVQA
LLaVA-Next	672x672	64.6	69.5	73.7
VLSA	672x672	65.3	71.4	75.2
LLaVA-Next	4096x4096	68.4	72.7	76.2
(ex7) VLSA	4096x4096	**69.5**	**76.6**	**80.1**

W3:

In response to the reviewer's suggestions, we have supplemented the preliminary results on the BLINK dataset, which involves visual prompts as inputs, and the interleaved benchmarks DEMON to further demonstrate the generalization capability of our method. Specifically, for the BLINK dataset, we integrated VLSA with both the 7B and 13B versions of LLaVA 1.5, comparing our outcomes against the officially reported results. For the DEMON benchmarks, we combined VLSA with LLaMA-Adapter V2 and LLAVA V1.0, once again contrasting our findings with the official benchmarks. Detailed implementation specifics will be included in the revised manuscript. The experimental results consistently demonstrate performance improvements achieved by our method, thereby validating its versatility.

Comparisions on DEMON:

Method	Multimodal Dialogue	Visual Storytelling	Visual Relation Inference	Multimodal Cloze	Knowledge Grounded QA	Text-Rich Image QA	Multi-Image Reasoning
LLaMA-Adapter V2	14.2	17.5	13.5	18.0	**44.8**	32.0	44.0
(ex8) LLaMA-Adapter V2 + VLSA	**16.0**	**17.9**	15.7	**19.2**	44.7	36.3	**45.5**
LLaVA	7.8	10.7	8.3	15.9	36.2	28.3	41.5
(ex9)LLaVA + VLSA	10.2	11.5	**15.8**	16.1	36.3	**37.2**	44.4

Comparisions on BLINK:

Method	Validation	Test
LLaVA-1.5 7B	37.1	38.0
(ex10) LLaVA-1.5 7B + VLSA	39.3	39.9
LLaVA-1.5 13B	42.7	40.6
(ex11) LLaVA-1.5 13B + VLSA	**46.1**	**45.3**