BEYOND MULTI-MODAL ALIGNMENT: AN INFORMATION-PRESERVING FRAMEWORK FOR VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Prevalent Vision-Language (VL) alignment techniques in Multi-modal Large Language Models (MLLMs) struggle to adequately align the language model with visual inputs, leading to hallucinations and undermining reliability. We rethink the modality alignment in MLLMs from the perspective of reducing information loss and present an efficient plug-in, VL Superior Alignment (VLSA), that decouples alignment into two stages. The first stage, **Perception Alignment**, couples compressive high-resolution encoding with reconstructive training through a pre-trained text-to-image latent diffusion model (LDM), encouraging compact visual embeddings to preserve input content and align with language. The second stage, **Cognition Alignment**, introduces auxiliary codebook-index prediction, supplemented by pixel-value prediction, to train the LLM's understanding and use of visual information. Analyses of these two stages yield two empirical insights: reconstruction fidelity alone does not fully characterize the usefulness of compact visual representations for language tasks, and the LLM's utilization of visual information can improve even when the representations are held fixed. Extensive experiments across over 30 MLLM benchmarks and 9 MLLM architectures, thorough ablations, and analyses of computational overhead demonstrate that VLSA consistently improves performance over high-resolution baselines while maintaining inference costs close to those of low-resolution models, with further gains when the computational budget is increased. In service to the MLLM research community, our code and model checkpoints will be publicly available.

1 INTRODUCTION

Large language models (LLMs) are advanced tools for understanding and generating contextual information. In order to transform LLMs into powerful multi-modal LLMs (MLLMs), current techniques (Bai et al., 2023; Li et al., 2023d; Gao et al., 2023; Liu et al., 2023b) generally adhere to standard processes involving using a pre-trained image encoder to embed visual context, then integrating the visual and textual embeddings into LLMs for a range of tasks. The feasibility of this paradigm hinges on the premise that LLMs can process visual embeddings in the same way they handle textual embeddings. Existing approaches like LLaVA (Liu et al., 2023b), PaLI (Chen et al., 2022), and CogVLM (Wang et al., 2024b) employ a linear layer or MLP as the projector to bridge the gap among embeddings, while models such as BLIP2 (Li et al., 2023d), InstructBLIP (Dai et al., 2023), and Qwen-VL (Bai et al., 2023) leverage Q-former (also named perceiver), which transfers arbitrary visual sequences into a fixed-length query through cross-attention, to achieve a similar effect. However, these Vision-Language (VL) alignment techniques primarily concentrate on achieving distributional or semantic consistency between VL embeddings, often overlooking the information loss during visual inference. Consequently, they may fail to adequately align the language model with visual inputs, leading to hallucinations and diminished reliability.

When contrasted to the ideal visual inference process illustrated at the top of Fig. 1(Left), which involves optimal visual encoding and response generation, actual visual inference empowered by conventional alignments shown at the bottom of Fig. 1(Left) highlights three distinct categories of information degradation that undermine system performance: (1) Original Information Loss (**O-IL**) stems from the limitations imposed by input resolution, where spatial downsampling of raw images inevitably discards critical visual details. (2) Encoding Information Loss (**E-IL**) arises from inherent limitations in current visual encoding architectures: both vision encoders and alignment projection modules function as lossy translators or compressors, compromising the information perceived by LLMs. (3) Decoding Information Loss (**D-IL**) occurs during the response generation, where LLMs

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

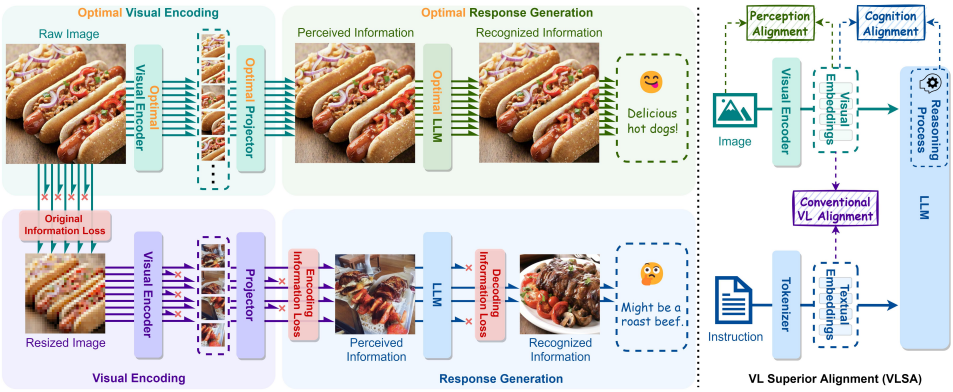

Figure 1: **Left:** Conventional Vision-Language (VL) alignments in MLLMs overlook information loss (IL) during visual inference. **Right:** To address this issue and further promote the alignment between the LLM and visual inputs, our VL Superior Alignment (VLSA) extends modality alignment in MLLMs beyond conventional VL alignments by incorporating Perception and Cognition Alignment.

demonstrate incomplete comprehension of encoded visual embeddings, causing the recognized visual information to diverge from what they perceived. Ultimately, D-IL leads to unreliable responses, even in scenarios with theoretically optimal visual encoding (i.e., when E-IL is absent).

Recent efforts (Liu et al., 2024a; Li et al., 2023a; Bai et al., 2023; Zhang et al., 2023b; Young et al., 2024) have mitigated the O-IL by preserving or increasing the resolution of input images rather than uniformly downsampling them during encoding. Typically, they involve segmenting a high-resolution image into patches, encoding each patch independently, and then concatenating them into a lengthy sequence that serves as visual embeddings. However, the efficiency of projector-based alignments (e.g., LLaVA) is diminished in this scenario, as LLMs' time complexity scales quadratically with input sequence length. Besides, they rely on the LLM to interpret relationships among image patches. Nevertheless, the causal attention inherent in LLMs limits their ability to accurately model these interrelationships, exacerbating the D-IL. On the other hand, Q-former-based alignments (e.g., BLIP2) are lossy compressors that may overlook crucial features (see Appendix Fig. 6 and Tab. 18 (ex7)), intensifying the E-IL despite their potential to reduce visual sequence length for improved efficiency.

Moreover, most current techniques rely on supervised fine-tuning (SFT) to promote LLMs' understanding of visual embeddings. Nonetheless, commonly used tasks, such as image captioning, visual question answering, and optical character recognition, only help LLMs grasp certain aspects of visual semantics, potentially exacerbating the D-IL. Recent progresses (Zhang et al., 2023a; Wu et al., 2024; Li et al., 2024a; Lai et al., 2024) have broadened LLMs' comprehension by introducing additional objectives, such as visual grounding and segmentation, which can alleviate the D-IL to some extent. However, these supplementary tasks always rely on manual or semi-automatic annotation processes that involve human participation, and they still tend to focus on particular pre-defined attributes of visual semantics (Appendix A.3 provides ablations on these objectives).

These limitations call for modality alignment that preserves visual information during encoding and enables the LLM to use it effectively during response generation. We therefore propose VL Superior Alignment (VLSA), guided by two design principles (Fig. 1, Right). **Perception Alignment** seeks compact visual representations that preserve input content and are compatible with language, targeting O-IL and E-IL at low computational cost. **Cognition Alignment** treats the LLM's interpretation and use of that content as an explicit objective of training and evaluation, targeting D-IL (see Appendix A.4 for both stages and their components). As depicted in Fig. 2, **Perception Alignment** involves compressive encoding for high-resolution images and a novel reconstructive training method inspired by the denoising process of Latent Diffusion Models (LDM) (Esser et al., 2024). The former compresses high-resolution information into the embeddings of downsampled images via SA-perceiver, a cross-attention-based module. While mitigating O-IL, it overcomes computational overheads of prior high/mixed-resolution methods and aims to alleviate D-IL and E-IL compared to projector-based and Q-former-based alignments, respectively. The latter requires the model to reconstruct inputs from random noise (with outputs of compressive encoding as cues), enabling the image encoder to capture more details and allowing the SA-perceiver to retain more input-specific information during compression. Concurrently, **Cognition Alignment** includes novel self-supervised

fine-tuning (SSFT) objectives that augment the standard SFT objective, simultaneously facilitating the LLM's comprehensive understanding of high-level and low-level visual semantics. These auxiliary objectives leverage the codebook indices from a frozen VQ-VAE (van den Oord et al., 2017) as discrete labels representing high-level semantics of images, while employing pixel values to convey low-level semantics. The MLLM is tasked with simultaneously predicting responses to instructions, codebook indices, and image pixel values. To summarize:

- We introduce an information-loss-aware perspective on multimodal alignment, revealing that existing approaches overlook information loss at multiple stages. Building on this insight, we broaden the scope of modality alignment to encompass Perception and Cognition Alignment, with the goal of jointly mitigating information loss throughout visual reasoning.
- Guided by this perspective, we propose VLSA, an efficient plug-in with two novel training mechanisms. Perception Alignment couples compressive high-resolution encoding with LDM-based reconstructive training that supervises visual embeddings before they enter the LLM. Cognition Alignment introduces codebook-index prediction as an auxiliary objective for visual understanding, supplemented by pixel-value prediction.
- Controlled analyses yield empirical insights that guide the design of visual compression and LLM supervision. Extensive experiments across over 30 benchmarks and 9 diverse MLLM architectures demonstrate consistent gains over high-resolution baselines while largely retaining the computational efficiency of low-resolution inference.

2 Related Work

Some recent refinements of MLLMs can be interpreted through the lens of reducing information loss.

Multi-resolution visual features. Works such as Monkey (Li et al., 2023f), LLaVA-UHD (Xu et al., 2024), and SliME (Zhang et al., 2024b) leverage both low and high-res visual features to improve performance, effectively reducing O-IL with reasonable computational efficiency. However, Monkey and LLaVA-UHD employ Q-former-like resamplers to reduce visual tokens, which, despite improving efficiency, may exacerbate E-IL, resulting in slower convergence and suboptimal performance (Fig. 6, Tab. 18(ex7) in Appendix). Similar to our approach, SliME uses low-resolution images as reasoning foundations and high-resolution patches as supplements. However, SliME implements a complex MOE adapter with a compression layer and a prompt-based router for crucial high-res tokens, requiring three-stage alternating training and additional data. In contrast, our VLSA comprehensively mitigates information loss while enhancing efficiency when processing multi-resolution inputs (see Appendix A.4.3 for architectural comparisons of SA-Perceiver with Q-Former and Mini-Gemini).

Multiple visual encoders. Approaches like BRAVE (Kar et al., 2024), COMM (Jiang et al., 2024), and LLaVA-Read (Zhang et al., 2024a) leverage multiple visual experts to reduce E-IL. Despite performance gains, this introduces significant computational overhead. Our VLSA simultaneously utilizes CLIP and VQ-VAE encoders, with VQ-VAE serving exclusively as an annotation tool to generate training data for SSFT objectives within Cognition Alignment. The entire VQ-VAE annotation process can occur offline, preserving computational efficiency.

Reconstructive training objectives. Advances like X-former (Swetha et al., 2024), LaViT (Jin et al., 2024), and Ross (Wang et al., 2024a) explore reconstructive training to reduce information loss. X-former employs MAE to recover local details missing from CLIP's global features, addressing encoder-related E-IL but leaving projector-induced E-IL. LaViT reconstructs outputs from its frozen visual encoder (not raw images), minimizing its vector quantization loss while failing to address encoder-related E-IL. Ross uses LLM-processed image tokens as reconstruction clues, effectively reducing E-IL but potentially interfering with the LLM's training and leading to suboptimal performance (please refer to Appendix A.12 for more analysis). Conversely, our VLSA comprehensively reduces information loss during visual perception without imposing the reconstruction objective on the LLM, while also promoting semantic alignment between VL features (see Appendix A.19 for a comparison of semantic alignment with and without LDM pretraining).

Vision-Language early fusion. Methods like QA-ViT (Ganz et al., 2024), EMMA (Ghazanfari et al., 2025), and mPLUG-Owl2 (Ye et al., 2023) enhance performance by modifying Visual-Language (VL) feature fusion locations, enabling preliminary VL interactions before features enter the LLM. Compared to standard approaches, early fusion mitigates loss of critical visual information by filtering visual semantics to focus on user requests, boosting performance on standard VQA tasks. However, this approach presents a trade-off: performance degrades with ambiguous or verbose user queries (please refer to Appendix A.11).

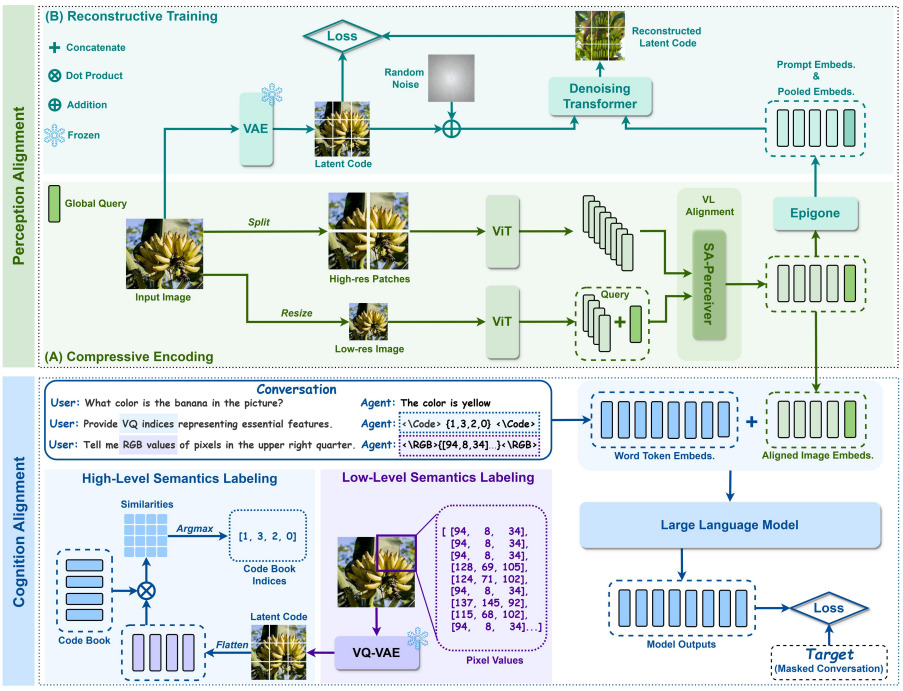

Figure 2: VLSA Architecture. TOP: Perception Alignment consists of (a) compressive encoding relieving O-IL while reducing system overhead, and (b) reconstructive training minimizing E-IL. Bottom: Cognition Alignment utilizes SSFT objectives to reduce D-IL.

3 Methodology

3.1 Perception Alignment

Compressive encoding for high-resolution images. VLSA takes an image X with arbitrary height and width as input. We minimally pad X to size $3 \times H \times W$, where H and W are divisible by the image encoder's input height h and width w, respectively. Following this, X is split into high-resolution patches $X_{\text{Hi}} \in \mathbb{R}^{m \times 3 \times h \times w}$, with m representing the number of patches. Simultaneously, X is downsampled to create a low-resolution snapshot $X_{\text{Lo}} \in \mathbb{R}^{3 \times h \times w}$. Both X_{Hi} and X_{Lo} are then separately encoded by the CLIP (Radford et al., 2021) vision encoder, resulting in visual embedding $V_{\text{Hi}} \in \mathbb{R}^{m \times l \times d}$ and $V_{\text{Lo}} \in \mathbb{R}^{l \times d}$, where l is the sequence length and d is the feature dimension. Differing from previous techniques that flatten multi-scale visual embeddings to a single sequence, we gather information from V_{Hi} while preserving their spatial structure according to V_{Lo}. This goal is achieved through the SA-Perceiver depicted in Fig. 5. Specifically, we append a learnable query $q \in \mathbb{R}^{1 \times d}$ to V_{Lo} to gather global visual features, resulting in $V'_{\text{Lo}} \in \mathbb{R}^{(l+1) \times d}$. Subsequently, we update V'_{Lo} with V_{Hi} by cross-attention:

$$Q = V'_{\text{Lo}} w_{\text{q}}, \quad K_{\text{c}} = V_{\text{c}} = \bar{V}_{\text{Hi}} w_{\text{kv}},$$
$$A_{\text{c}} = \text{softmax}\left(QK_{\text{c}}^{\top}/\sqrt{d}\right), \quad H = V'_{\text{Lo}} + \sigma((A_{\text{c}}V_{\text{c}})w_{\text{o1}}). \tag{1}$$

where $\bar{V}_{\text{Hi}} \in \mathbb{R}^{(ml) \times d}$ stacks the token embeddings from all m high-resolution patches in spatial order. The projection matrices w_{q}, w_{kv}, and w_{o1} have size $d \times d$, with w_{kv} shared by the key and value projections. σ denotes the SiLU activation function (Elfwing et al., 2018). We then apply self-attention to H to process the aggregated high-resolution information and refine the global representation:

$$A_{\text{s}} = \text{softmax}\left(H(Hw_{\text{k}})^{\top}/\sqrt{d}\right), \qquad Y = (A_{\text{s}}H)w_{\text{o2}}. \tag{2}$$

The key projection is $w_{\text{k}} \in \mathbb{R}^{d \times d}$. The output projection is $w_{\text{o2}} \in \mathbb{R}^{d \times d_{\text{LLM}}}$, where d_{LLM} denotes the LLM embedding dimension. Finally, we split Y into visual embeddings $V \in \mathbb{R}^{l \times d_{\text{LLM}}}$ and global embedding $P \in \mathbb{R}^{d_{\text{LLM}}}$ for the subsequent reconstructive training. Furthermore, the SA-Perceiver is also responsible for achieving VL alignment, similar to the role of the MLP projector in LLaVA or the Q-former in BLIP-2. Appendices A.4.1 and A.4.3 provide more design rationale and insights.

Compared with MLP projectors, our method mitigates the challenge of modeling visual content solely via causal attention in LLMs, reducing D-IL risk while lowering computational overhead by shortening visual embedding sequences. Meanwhile, compared to Q-former, SA-perceiver uses downsampled images rather than learnable vectors as queries to aggregate high-resolution information, reducing the risk of losing critical visual cues and thus mitigating E-IL (please refer to Appendix Fig. 6 and Tab. 18(ex7) for more details).

Reconstructive training with LDM. To minimize information loss during image encoding, we encourage visual embeddings to retain information needed to reconstruct the input image X. Drawing inspiration from Latent Diffusion Models (LDM), we initiate this reconstruction from Gaussian noise, with V and P serving as vital clues. As illustrated in Fig. 2 (B), the process begins with encoding X into the latent code $z \in \mathbb{R}^{s \times d_z}$ through a frozen Variational Auto-Encoder (VAE) (Kingma, 2013), where s and d_z denote the latent sequence length and feature dimension, respectively. Following this, we add Gaussian noise onto z, the intensity of which depends on the time step t, yielding a noisy latent z_t. We train a denoising transformer to reconstruct z from z_t using V and P as guidance (Appendix A.3 shows single-step denoising is sufficient). Both VAE and the denoising transformer are initialized from the text-to-image LDM, Stable Diffusion 3-medium (SD) (Esser et al., 2024). The reconstruction loss is formulated as:

$$\mathcal{L}_{\text{rec}} = \| z - z_\theta \left(z_t,\ c \right) \|_2^2, \tag{3}$$

where $z_\theta(z_t, c)$ is the denoising transformer's prediction with learnable parameters θ, and c is its conditioning input, defined by

$$c = \text{Epigone}(\text{concat}(V, P)) \,. \tag{4}$$

The Epigone maps V and P to the prompt and pooled embeddings required by SD. Unlike the word embedding layer in SD, which conveys natural language's concise high-level semantics, the Epigone is tailored to transmit detailed image information. As a result, it mitigates the randomness (detrimental to reconstruction) inherent in SD. To enhance computational efficiency, we adopt a simple MLP as the Epigone.

The key advantage of a pretrained text-to-image LDM over a conventional autoencoder is that it can promote vision–language alignment alongside information preservation. Our reconstructive objective exploits the LDM's learned correspondence between language and images to supervise visual embeddings before LLM processing, with reconstruction gradients updating both the vision encoder and alignment projector (see Appendix A.4.2 for further analysis, and Appendices A.3 and A.19 for CKA comparisons with MAE-based reconstruction and SD3 trained from scratch, respectively).

3.2 Cognition Alignment

Auxiliary self-supervised objectives. To reduce information loss during response generation, we introduce codebook-index prediction as an auxiliary self-supervised fine-tuning (SSFT) objective for the LLM. It requires the LLM to infer local visual content and spatial structure from the embeddings used for instruction following. Pixel-value prediction supplies additional supervision of low-level appearance, and both objectives augment standard supervised fine-tuning (SFT). To discretely label high-level semantics in a self-supervised manner, we encode image X into latent code $L \in \mathbb{R}^{l' \times d'}$ using a frozen VQ-VAE (Vector Quantized VAE) (van den Oord et al., 2017) initialized from (Tang et al., 2023). Its codebook is denoted by $B \in \mathbb{R}^{b \times d'}$. Here, l' is the latent sequence length, b is the codebook size, and d' is the latent dimension. We precompute the fixed high-level target sequence $Target_{\text{VQ}} \in \{0, \ldots, b-1\}^{l'}$ by assigning each latent vector the index of its nearest codebook entry:

$$(Target_{\text{VQ}})_i = \underset{j \in \{0,\ldots,b-1\}}{\arg\min} \|L_i - B_j\|_2^2, \qquad i = 0, \ldots, l' - 1. \tag{5}$$

For low-level semantics, the target $Target_{\text{PX}} \in \mathbb{R}^{3 \times h \times w}$ consists of the RGB pixel values of X_{Lo} (Sec. 3.1). Our SSFT objectives require the LLM to predict $Target_{\text{VQ}}$ and $Target_{\text{PX}}$ from visual embeddings V in Sec 3.1. In practice, we modify $Target_{\text{PX}}$ to be pixel values within a randomly selected quarter of X_{Lo} to reduce the overhead. The 10:1 ratio between codebook and pixel prediction tasks is empirically optimized (see Appendix A.3).

Unlike previous attempts to enhance the understanding of visual semantics in specific categories, our approach offers a more holistic visual comprehension without manual annotation. This is

Table 1: **Comparisons on academic-task-oriented datasets**. VLSA uses the efficiency-oriented 336* setting with 576 visual tokens. †Includes in-house data that is not accessible. Res, PT, and IT indicate image resolution and the number of samples in pretraining and finetuning, respectively.

Method	LLM	Res.	PT	IT	VQA^{v2}	GQA	VisWiz	COCO	TextC	VQAST	SQA^{I}	SQA	VQA^{T}
BLIP-2	Vicuna-13B	224	129M	-	41.0	41	19.6	–	–	36.4	61	-	42.5
InstructBLIP	Vicuna-13B	224	129M	1.2M	–	49.5	33.4	–	–	38.7	63.1	-	50.7
Shikra	Vicuna-13B	224	600K	5.5M	77.4	–	–	–	–	–	–	-	–
IDEFICS-80B	LLaMA-65B	224	353M	1M	60.0	45.2	36.0	–	56.8	–	–	-	30.9
Qwen-VL-Chat	Qwen-7B	448	1.4B†	50M†	78.2	57.5	38.9	–	–	–	68.2	-	**61.5**
LLaVA-1.5	Vicuna-7B	336	558K	665K	76.6	62.0	50.0	109.4	101.8	54.0	69.8	70.0	58.2
LLaVA-1.5-HD	Vicuna-7B	Any	558K	665K	81.2	64.6	54.1	125.4	107.3	63.9	71.2	71.8	62.1
+VLSA	Vicuna-7B	336*	558K	665K	**83.0**	**65.2**	**61.5**	**129.3**	**107.5**	**65.0**	**73.9**	**76.5**	**63.0**
LLaVA-Next	LLaMA3-8B	Any	558K	790K	82.4	64.9	46.7	137.3	70.1	64.2	74.6	72.1	63.9
+VLSA	LLaMA3-8B	336*	558K	790K	**83.5**	**65.3**	**57.7**	**139.5**	**73.3**	**65.7**	**77.5**	**78.6**	**65.2**

because the VQ-VAE we employed functions as an autoencoder, with its codebook indices capable of encapsulating a broad spectrum of semantics that transcend specific, human-defined categories.

Our intuition behind the effectiveness of predicting codebook indices: During SFT, the LLM partially understands the mapping between visual semantics and natural language. Meanwhile, it grasps the majority of the mappings between visual semantics and codebook indices during our SSFT. Given the overlap in the visual semantics present in both mapping sets, the LLM is able to establish connections between natural language and codebook indices. Consequently, SSFT indirectly enhances the LLM's ability to map visual semantics to natural language. Please refer to Appendices A.7 and A.4.4 for more analysis. Appendix A.3 demonstrates robustness across different quantized encoders.

Joint training with multiple objectives. During visual instruction tuning, we augment the training conversations with codebook-index and pixel-value prediction pairs and optimize the resulting responses using the standard language-modeling loss $\mathcal{L}$, following LLaVA (Liu et al., 2023b). Combined with reconstructive training, the overall objective is $\mathcal{L}_{\text{all}} = \mathcal{L} + \mathcal{L}_{\text{rec}}$. Appendices A.9 and A.13 provide the optimization details, instruction templates, and data-construction procedure.

4 Experiments

We assess VLSA by integrating it into prevalent MLLM architectures MiniGemini (Li et al., 2024b), Qwen2.5-VL (Bai et al., 2025), InternVL-2.5 (Chen et al., 2025b), LLaVA-UHD (Xu et al., 2024), SliME (Zhang et al., 2024b), LLaVA-1.5-HD (Liu et al., 2023a), LLaVA-Next (Liu et al., 2024a), LaViT (Jin et al., 2024), and Ovis (Lu et al., 2024). The LLaVA series serves as the primary setting for detailed analysis because its simple MLP projector and standard SFT provide a clear baseline for evaluating VLSA's components. Appendix A.2 reports the model configurations, training data, and two-stage optimization schedule.

Efficiency-oriented setting. Unless otherwise specified, all LLaVA-based comparisons use our default 336* configuration. Encoding the 336×336 snapshot X_{Lo} with CLIP-ViT-L/14@336px yields $l = 576$ spatial query tokens (Appendix A.2). The SA-Perceiver preserves this sequence length when aggregating V_{Hi}, so the visual sequence V passed to the LLM also contains 576 tokens. This setting maintains inference costs comparable to low-resolution processing.The main results evaluate VLSA under this efficiency constraint, while Table (5) examines its performance when the snapshot resolution and visual-token budget are increased.

4.1 Experimental results

Quantitative Results. Tables 1 and 2 compare VLSA with representative MLLMs on eight academic-task-oriented datasets and seven instruction-following benchmarks. Across the paired LLaVA-1.5-HD and LLaVA-Next comparisons, VLSA improves most metrics, with particularly large gains on VizWiz (+7.4/+11.0), ScienceQA-IMG (+4.7/+6.5), MME (+117/+91), SEED-Bench (+5.8/+5.6), and MM-Vet (+4.7/+6.3). VQA^{v2}, GQA, and COCO show smaller gains, possibly because images or annotations from these datasets are already included in the LLaVA fine-tuning data, leaving less room for improvement over the matched baselines. Appendix A.1.1 provides the benchmark definitions and complete task-level analysis.

Visual-heavy and hallucination evaluations. On visual-heavy benchmarks, VLSA outperforms the Any-resolution LLaVA-Next baseline on the tasks reported in Table 4 and Appendix A.1.3; it also improves V*Bench from 48.68 to 52.77 and MMVP from 24.70 to 29.60 against a matched LLaVA-1.5 baseline (Appendix A.20). On hallucination benchmarks, Table 6 shows consistent

Table 2: **Comparison for multi-modal comprehension on MLLM benchmarks**. VLSA uses the efficiency-oriented 336* setting with 576 visual tokens. ‡Evaluating via text-only GPT-4-0613.

Method	LLM	Res.	PT	IT	POPE	MME	MMB	MMB^{CN}	SEED	LLaVA^{W}	MM-Vet
BLIP-2	Vicuna-13B	224	129M	-	85.3	1293.8	–	–	46.4	38.1‡	22.4‡
InstructBLIP	Vicuna-13B	224	129M	1.2M	78.9	1212.8	–	–	–	58.2‡	25.6‡
Shikra	Vicuna-13B	224	600K	5.5M	–	–	58.8	–	–	–	–
IDEFICS-80B	LLaMA-65B	224	353M	1M	–	–	54.5	38.1	–	–	–
Qwen-VL-Chat	Qwen-7B	448	1.4B†	50M†	–	1487.5	60.6	56.7	58.2	–	–
LLaVA-1.5	Vicuna-7B	336	558K	665K	85.9	1462.6	64.8	57.6	58.6	63.2‡	30.6‡
LLaVA-1.5-HD	Vicuna-7B	Any	558K	665K	87.1	1505	67.5	64.0	61.3	69.5‡	31.2‡
+VLSA	Vicuna-7B	336*	558K	665K	**87.8**	**1622**	**68.3**	**64.8**	**67.1**	**70.5**‡	**35.9**
LLaVA-Next	LLaMA3-8B	Any	558K	790K	87.7	1753	**72.2**	70.1	60.0	69.4‡	33.4‡
+VLSA	LLaMA3-8B	336*	558K	790K	**88.6**	**1844**	71.9	**71.5**	**65.6**	**72.1**‡	**39.7**‡

improvements on Hallusion-Bench and AMBER-Discriminative across both LLaVA backbones, complementing the POPE results in Table 2.

Table 3: **Generalizability of VLSA to more MLLM architectures.** Each baseline/+VLSA pair uses matched initialization, training data, and evaluation.

Method	LLM	GQA	VQA^{T}	MMB	MME	MM-Vet	LLaVA^{W}	MMMU$_{val}$
MiniGemini-HD	Vicuna-7B	66.4	68.4	**65.8**	1865	41.3	63.0	36.8
+VLSA	Vicuna-7B	**66.9**	**69.5**	65.6	**1944**	**49.2**	**65.2**	**37.3**
Qwen2.5-VL	Qwen2.5-7B	68.7	84.1	82.8	2303	62.5	74.7	58.6
+VLSA	Qwen2.5-7B	**69.0**	**86.2**	**85.8**	**2354**	**66.3**	**74.8**	**61.4**
InternVL-2.5	InternLM2.5-7B	78.2	77.4	84.3	2322	64.9	78.5	53.8
+VLSA	InternLM2.5-7B	78.2	**79.6**	**84.8**	**2358**	**66.1**	**79.4**	**56.1**
LLaVA-UHD	Vicuna-13B	65.3	65.0	67.8	1535	33.8	64.2	37.1
+VLSA	Vicuna-13B	**65.7**	**66.1**	**68.2**	**1573**	**35.6**	**65.7**	**37.4**
SliME	LLaMA3-8B	64.6	64.1	70.9	1861	35.4	70.2	42.6
+VLSA	LLaMA3-8B	**65.3**	**64.9**	**74.6**	**1887**	**38.0**	**70.8**	**46.0**

Cross-architecture generalization and stability. Table 3 reports matched baseline/+VLSA comparisons across five additional MLLM architectures. Each pair shares the same initialization, training data, and evaluation protocol, and thus provides the basis for assessing the effect of VLSA. Appendix A.18 separately reports the results published for the original models as contextual references, since their training or evaluation settings may differ from ours. Appendix A.8 further evaluates compatibility with LaViT and Ovis. Run-to-run stability on representative benchmarks is reported in Appendix A.3.

Qualitative Results. To underscore the efficacy of our perception alignment, we conduct an analytical visualization comparing reconstructed images based on visual embeddings from LLaVA-Next and our VLSA. As shown in Appendix Fig. 4, the baseline trends overlook substantial semantics in the original images. By contrast, our VLSA with reconstructive training minimizes detail loss (Appendix A.1.4 provides quantitative analysis of reconstructed images). To further demonstrate the advantages of our perception and cognition alignment, we provide additional results on visual writing and visual question-answering tasks in Appendix A.14.

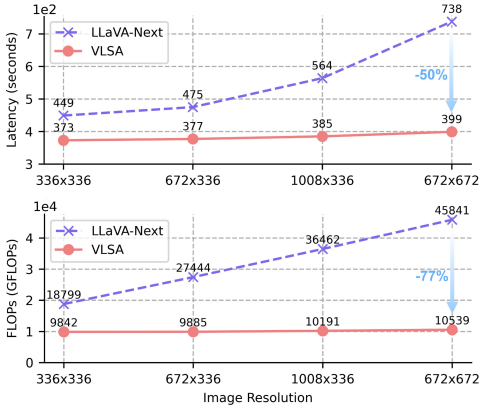

Figure 3: Latency and FLOPs in processing 1,000 images under various resolutions. VLSA keeps its 336* snapshot/token budget.

Efficiency Evaluation. To evaluate VLSA's computational efficiency, we randomly select 1,000 instances from the 790K instruction-tuning set, remove the textual instructions, resize the images to different resolutions, and compare latency and FLOPs on a single NVIDIA A100 GPU with a batch size of 1. As shown in Fig. 3, despite introducing multi-scale feature interactions in SA-Perceiver, VLSA processes high-resolution inputs at inference costs comparable to low-resolution processing and substantially lower than the high-resolution baseline. Throughout this evaluation, VLSA retains its 336* snapshot and token budget. Further analysis is provided in Appendix A.5.

Table 4: **Results on document understanding tasks and ablation studies.** Numbers in parentheses denote (x/y), where x and y are performance changes relative to baseline (1) and (2).

Method	Res.	AI2D	ChartQA	DocVQA	MME^C	MME^P	SQA	VisWiz
Baseline Performances								
(1) LLaVA-Next	Any	69.5 (+0/+1.7)	67.1 (+0/+22.3)	73.7 (+0/+28.3)	298.9 (+0/-23.2)	1455.7 (+0/+38.4)	72.1 (+0/-6.2)	46.7 (+0/-2.2)
(2) LLaVA-Next	336	67.8 (-1.7/+0)	44.8 (-22.3/+0)	45.4 (-28.3/+0)	322.1 (+23.2/+0)	1417.3 (-38.4/+0)	78.3 (+6.2/+0)	48.9 (+2.2/+0)
Ablations on the Perception Alignment								
(3) LLaVA-Next + (a)	336^*	69.7 (+0.2/+1.9)	65.3 (-1.8/+20.5)	61.8 (-11.9/+16.4)	315.0 (+16.1/-7.1)	1426.5 (-29.2/+9.2)	78.1 (+6.0/-0.2)	55.9 (+9.2/+7.0)
(4) LLaVA-Next + (ab)	336^*	68.2 (-1.3/+0.4)	67.4 (+0.3/+22.6)	**75.5** (+1.8/+30.1)	329.9 (+31.0/+7.8)	1481.4 (+25.7/+64.1)	74.1 (+2.0/-4.2)	53.6 (+6.9/+4.7)
Ablations on the Cognition Alignment								
(5) LLaVA-Next + (c)	Any	**72.5** (+3.0/+4.7)	67.0 (-0.1/+22.2)	73.9 (+0.2/+28.5)	**353.6** (+54.7/+31.5)	**1570.0** (+114.3/+152.7)	72.6 (+0.5/-5.7)	50.9 (+4.2/+2.0)
(6) LLaVA-Next + (abc)	336^*	71.4 (+1.9/+3.6)	**67.9** (+0.8/+23.1)	75.2 (+1.5/+29.8)	336.4 (+37.5/+14.3)	1507.4 (+51.7/+90.1)	**78.6** (+6.5/+0.3)	**57.7** (+11.0/+8.8)

Table 5: **Resolution scaling.** Any-resolution baseline and two VLSA snapshot resolutions.

Method	Res.	GQA	SQA-I	DocVQA
LLaVA-Next	Any	64.9	74.6	73.7
VLSA	336^*	65.3	77.5	75.2
VLSA	672	**69.2**	**80.8**	**80.1**

Table 6: **Hallucination benchmarks.**

Backbone	Variant	Hallusion	AMBER
LLaVA-1.5-HD	Baseline	44.4	75.3
LLaVA-1.5-HD	+VLSA	**49.2**	**82.5**
LLaVA-Next	Baseline	44.2	78.7
LLaVA-Next	+VLSA	**48.1**	**83.2**

Despite the additional computation introduced by LDM-based reconstructive training, VLSA reduces the total wall-clock training time of LLaVA-Next from 35.3 to 18.9 hours (-47%), because its shorter visual sequence substantially lowers LLM-side computation. As reconstructive training only provides auxiliary supervision during training, the VAE, denoising transformer, and Epigone are not retained for inference. Appendix A.15 provides the detailed breakdown.

Relaxing the efficiency constraint. We next increase the snapshot resolution to 672×672, allowing a larger visual-token budget. Table 5 shows further gains on GQA, SQA-I, and DocVQA, demonstrating the performance headroom available beyond the efficient default setting.

4.2 Ablation Study

We ablate (a) compressive encoding, (b) reconstructive training, and (c) Cognition Alignment using six LLaVA-Next variants (Table 4): (1) the baseline, (2) reduced resolution, (3) adding (a), (4) adding (a) and (b) for Perception Alignment, (5) adding only (c), and (6) full VLSA. Seven metrics cover perception (ChartQA, DocVQA, and MME-perception), cognition (MME-cognition), or both (AI2D, SQA, and VizWiz).

Effectiveness of Perception Alignment. Tab. 4 (1) and (2) indicate that reducing the input resolution severely impairs perceptual capabilities. However, (2) outperforms baseline (1) on MME-cognition, SQA, and VisWiz, revealing that previous techniques do not effectively model high-resolution inputs. Through (1), (2), and (3), we observe that (a) alleviates the perceptual challenges posed by low-resolution inputs while retaining the benefits of (2). The improvement from (2) to (3) primarily reflects recovery from the additional O-IL introduced by downsampling. However, (3) remains below the high-resolution baseline (1) on ChartQA, DocVQA, and MME-perception. Adding reconstructive training in (4) raises all three above (1) without increasing the visual-token budget, supporting its role in mitigating E-IL during compression. The choice of reconstruction supervision also matters. Compared with reconstructive training using a masked autoencoder (MAE), LDM-based training yields higher GQA and SQA-I scores and greater similarity between visual and language representations in most tested LLM layers, despite comparable reconstruction fidelity (Appendix Tables 19 and 29). These findings support coupling visual compression with supervision that promotes both input fidelity and language compatibility.

Effectiveness of Cognition Alignment. By comparing (6) with (4), we find that (c) improves most metrics, consistent with better understanding of both shallow and deep semantics of visual embeddings. Isolated testing of (c) in (5) also yields substantial gains on AI2D, MME-cognition, MME-perception, and VizWiz, while retaining the computational cost of the high-resolution baseline. Full VLSA (6) improves all seven metrics over (1) under the efficient 336^* setting. In addition, Table 7 disentangles the two SSFT objectives. Pixel-value prediction improves all three benchmarks when combined with codebook-index prediction, although it is not consistently beneficial when used alone. One possible explanation is that predicting exact RGB values introduces a dense low-level objective that may compete with language modeling. In the joint setting, codebook indices provide compressed discrete targets for local content and spatial structure, while pixel values supplement them with color,

Table 7: **Ablations on SSFT objectives.**

Pixel Values	Codebook indices	AI2D	SQA	ChartQA
✗	✗	68.2	74.1	67.4
✗	✓	69.8	77.2	67.7
✓	✗	68.4	71.6	67.5
✓	✓	**71.4**	**78.6**	**67.9**

Table 8: **Controlled proxy evaluations of E-IL and D-IL.**

Target loss	Proxy metric	Reference	Aligned
E-IL	Shared-decoder PSNR (dB)	12.2	**21.8**
D-IL	Matched-image score	-0.22	0.08
	Mean shuffled-image score	-0.80	-0.79
	Utilization gap U	0.58	0.87
	ΔU	$+0.29$ $[0.23, 0.35]$	

brightness, texture, and other appearance details not fully retained after quantization. Appendix A.21 separately evaluates the prediction fidelity of both objectives.

To complement the task-level ablations, we use Grad-CAM (Selvaraju et al., 2017) to visualize which image regions contribute to the LLM's answers and centered kernel alignment (CKA) (Kornblith et al., 2019) to compare visual and language representations. Compared with VLSA without Cognition Alignment, full VLSA exhibits more concentrated visual attention and greater similarity between visual and language representations at intermediate layers. Appendix A.16 provides the visualizations and detailed results across layers.

Controlled evaluation of E-IL and D-IL. We conduct two controlled experiments to assess information retention during visual encoding and visual-information utilization during response generation, corresponding to E-IL and D-IL, respectively.

For E-IL, we freeze the LLaVA-Next and VLSA visual pipelines, jointly train one decoder with shared conditioning projections on both representation types, and then hold it fixed while reconstructing the same unseen images. Because the decoder and evaluation images are shared, PSNR serves as a proxy for differences in the information recoverable from the visual embeddings. The higher shared-decoder PSNR of VLSA in Table 8 supports lower E-IL under this controlled setting. Appendix A.17 gives the complete protocol, while Appendix A.1.4 reports complementary reconstruction and frequency-domain analyses.

For D-IL, we compare LLMs trained with and without Cognition Alignment using identical visual representations extracted from the checkpoint without Cognition Alignment. For each question, we first evaluate both LLMs with the representation of its associated image. We then repeat the evaluation after randomly shuffling the representations across questions, ensuring that each question receives the representation of another image. The questions remain unchanged, and both LLMs use the same three shuffles. We define the utilization gap U as the matched-image score minus the mean shuffled-image score, and ΔU as the mean change in U with Cognition Alignment. Appendix A.17 provides the score definitions and full experimental protocol. Table 8 shows that Cognition Alignment increases the matched-image score while leaving the mean shuffled-image score nearly unchanged, yielding $\Delta U > 0$. This result shows improved use of image-specific information under fixed visual representations, providing controlled evidence of lower D-IL. It supports explicitly training and evaluating visual-information utilization alongside representation learning.

Additional evaluations and ablations. Appendix A.3 reports supplementary ablations on architecture, training objectives, resolution, and stability. Appendix A.6 examines CLIP-score and GQA learning curves, while Appendices A.1.5 and A.1.6 evaluate vision-only and text-only performance.

5 Discussion and Conclusion

VLSA realizes Perception Alignment through compressive encoding with LDM-based reconstructive training, and Cognition Alignment through auxiliary codebook-index and pixel-value prediction. Extensive experiments demonstrate consistent gains over high-resolution baselines at inference costs close to low-resolution processing. Beyond these performance gains, our analyses yield two empirical insights: 1) Reconstruction fidelity alone does not fully characterize compact representations' usefulness for language reasoning, and 2) the LLM can also improve its use of retained visual information under fixed representations. These insights suggest two directions for future work on efficient MLLMs: jointly designing compression and supervision for input fidelity and language compatibility, and explicitly training and evaluating the LLM's use of retained information. **Limitations & Future Work.** The current approach leverages multiple optimization objectives uniformly throughout training, achieving strong performance. A promising avenue for future research is to explore stage-adaptive strategies that dynamically adjust the emphasis on different objectives during training, potentially unlocking additional performance gains.

AI USE STATEMENT

We used generative AI tools to assist with language polishing and LaTeX typesetting, including table formatting and layout adjustments. The authors reviewed all AI-assisted revisions and take full responsibility for the final manuscript.

REFERENCES

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. doi: 10.1109/ICCV.2019.00904.

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.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4291–4301, 2019.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025a.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. arxiv 2015. *arXiv preprint arXiv:1504.00325*, 2015.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025b. URL https://arxiv.org/abs/2412.05271.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pp. 148–166. Springer, 2024.

Roy Ganz, Yair Kittenplon, Aviad Aberdam, Elad Ben Avraham, Oren Nuriel, Shai Mazor, and Ron Litman. Question aware vision transformer for multimodal reasoning. *arXiv preprint arXiv:2402.05472*, 2024.

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

Sara Ghazanfari, Alexandre Araujo, Prashanth Krishnamurthy, Siddharth Garg, and Farshad Khorrami. Emma: Efficient visual alignment in multi-modal llms, 2025. URL https://arxiv.org/abs/2410.02080.

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*, pp. 6904–6913, 2017.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–14385, 2024.

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

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

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*, pp. 6700–6709, 2019.

Dongsheng Jiang, Yuchen Liu, Songlin Liu, Jin'e Zhao, Hao Zhang, Zhen Gao, Xiaopeng Zhang, Jin Li, and Hongkai Xiong. From clip to dino: Visual encoders shout in multi-modal large language models, 2024. URL https://arxiv.org/abs/2310.08825.

Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Quzhe Huang, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, Di Zhang, Wenwu Ou, Kun Gai, and Yadong Mu. Unified language-vision pretraining in llm with dynamic discrete visual tokenization, 2024. URL https://arxiv.org/abs/2309.04669.

Oğuzhan Fatih Kar, Alessio Tonioni, Petra Poklukar, Achin Kulshrestha, Amir Zamir, and Federico Tombari. Brave: Broadening the visual encoding of vision-language models, 2024. URL https://arxiv.org/abs/2404.07204.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 235–251. Springer, 2016.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9579–9589, 2024.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. Obelics: An open web-scale filtered dataset of interleaved image-text documents, 2023.

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*, 2023a.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023b.

Juncheng Li, Kaihang Pan, Zhiqi Ge, Minghe Gao, Wei Ji, Wenqiao Zhang, Tat-Seng Chua, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Fine-tuning multimodal llms to follow zero-shot demonstrative instructions. *arXiv preprint arXiv:2308.04152*, 2023c.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023d.

Rongjie Li, Yu Wu, and Xuming He. Learning by correction: Efficient tuning task for zero-shot generative vision-language reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13428–13437, 2024a.

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*, 2024b.

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*, 2023e.

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

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023b.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 1 2024a. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

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*, 2023c.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12), December 2024b. ISSN 1869-1919. doi: 10.1007/s11432-024-4235-6. URL http://dx.doi.org/10.1007/s11432-024-4235-6.

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.

Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model, 2024. URL https://arxiv.org/abs/2405.20797.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning, 2022.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

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*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 742–758. Springer, 2020.

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*, pp. 8317–8326, 2019.

Sirnam Swetha, Jinyu Yang, Tal Neiman, Mamshad Nayeem Rizve, Son Tran, Benjamin Yao, Trishul Chilimbi, and Mubarak Shah. X-former: Unifying contrastive and reconstruction learning for mllms, 2024. URL https://arxiv.org/abs/2407.13851.

Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models, 2023. URL https://arxiv.org/abs/2205.16007.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024a.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9568–9578, 2024b.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Haochen Wang, Anlin Zheng, Yucheng Zhao, Tiancai Wang, Zheng Ge, Xiangyu Zhang, and Zhaoxiang Zhang. Reconstructive visual instruction tuning, 2024a. URL https://arxiv.org/abs/2410.09575.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, et al. Amber: An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models, 2024b.

Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Chunyi Li, Wenxiu Sun, Qiong Yan, Guangtao Zhai, et al. Q-bench: A benchmark for general-purpose foundation models on low-level vision. *arXiv preprint arXiv:2309.14181*, 2023.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024. URL https://arxiv.org/abs/2312.14135.

Size Wu, Sheng Jin, Wenwei Zhang, Lumin Xu, Wentao Liu, Wei Li, and Chen Change Loy. F-lmm: Grounding frozen large multimodal models. *arXiv preprint arXiv:2406.05821*, 2024.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, Maosong Sun, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images, 2024. URL https://arxiv.org/abs/2403.11703.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration, 2023. URL https://arxiv.org/abs/2311.04257.

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.

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.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and Huan Sun. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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*, pp. 11975–11986, 2023.

Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Lei Zhang, Chunyuan Li, and Jianwei Yang. Llava-grounding: Grounded visual chat with large multimodal models, 2023a. URL https://arxiv.org/abs/2312.02949.

Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023b.

Ruiyi Zhang, Yufan Zhou, Jian Chen, Jiuxiang Gu, Changyou Chen, and Tong Sun. Llava-read: Enhancing reading ability of multimodal language models, 2024a. URL https://arxiv.org/abs/2407.19185.

Yi-Fan Zhang, Qingsong Wen, Chaoyou Fu, Xue Wang, Zhang Zhang, Liang Wang, and Rong Jin. Beyond llava-hd: Diving into high-resolution large multimodal models, 2024b. URL https://arxiv.org/abs/2406.08487.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024c.

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

APPENDIX CONTENTS

A Appendix **17**

A.1 More Experimental Results and Quantitative Analysis 17
A.2 Implementation Details 21
A.3 More Ablations 22
A.4 Design Rationale and Technical Details 27
A.5 Analysis of Efficiency Evaluation 29
A.6 More Evaluations on Modality Alignment 29
A.7 The Effectiveness of Predicting Codebook Indices 30
A.8 More Model Integration Studies 30
A.9 Language Modeling Loss 31
A.10 Comparison with Native Multimodal Frameworks 32
A.11 Comparison with Early fusion Methods 32
A.12 Comparison with existing Reconstructive Training Objectives. 33
A.13 Cognition Alignment Templates 33
A.14 More Case Studies 34
A.15 Training Efficiency Analysis 37
A.16 Attention and Representation Diagnostics 37
A.17 Controlled Recoverability and Utilization Protocols 38
A.18 Matched Baselines and Original Model Context 41
A.19 Effect of Reconstruction Initialization 41
A.20 Vision-Centric Evaluation 42
A.21 Fidelity of Auxiliary Prediction Targets 42

A Appendix

A.1 More Experimental Results and Quantitative Analysis

A.1.1 Detailed Analysis of Main-Text Results

This section provides the complete analyses underlying the main-text results, including task-level interpretation, matched-setting comparisons, and run-to-run stability.

Academic-task and instruction-following results.

To demonstrate the effectiveness of VLSA, we first integrate it into the LLaVA series and compare the performances with BLIP-2 (Li et al., 2023d), InstructBLIP (Dai et al., 2023), Shikra (Chen et al., 2023), IDEFICS (Laurençon et al., 2023), and Qwen-VL (Bai et al., 2023).

Tab. 1 reports performances on 8 academic-task-oriented datasets (each examines a specific ability), including 6 VQA tasks: VQA^{v2} (Goyal et al., 2017), GQA (Hudson & Manning, 2019), VisWiz (Gurari et al., 2018), ST-VQA (Biten et al., 2019), ScienceQA-IMG (Lu et al., 2022), TextVQA (Singh et al., 2019) and 2 image captioning tasks: TextCaps (Sidorov et al., 2020), COCO (Chen et al., 2015). The table shows that VLSA demonstrates a significant and consistent improvement compared to the baselines. Specifically, VLSA notably enhances performance on unseen datasets VisWiz (+7.4/+11.0) and ScienceQA (+4.7/+6.5), which are not included in the training sets. The following are analyses of this phenomenon:

VizWiz's distinctive feature lies in the images captured by visually impaired users, which often exhibit heavy blurring, poor lighting, and incomplete content. Baselines may struggle to parse such degraded visual inputs, leading to hallucinations during testing. In contrast, VLSA, through Perception Alignment, better preserves valuable visual semantics from low-quality images during encoding, thereby enhancing the reliability of question answering. Besides, SQA's characteristic lies in questions often have lengthy lectures or hints to guide logical reasoning. However, verbose contexts might interfere with baseline models in interpreting and leveraging visual embeddings. VLSA alleviates this issue via Cognition Alignment, which mitigates the LLM's tendency to misinterpret or overlook semantics in visual embeddings within long-context scenarios.

However, VLSA does not lead to a notable performance boost on VQA^{v2}, GQA, and COCO. This observation may be attributed to the inclusion of images or annotations from these datasets during the fine-tuning process, utilizing 665K/790K data from LLaVA. Consequently, the baseline model also performs well on these datasets.

Besides, we further assess VLSA's multi-modal comprehension ability on 7 instruction-following benchmarks that are tailor-made for MLLMs, including POPE (Li et al., 2023e), MME (Fu et al., 2023), MMBench (Liu et al., 2023c), MMBench-Chinese (Liu et al., 2023c), SEED-Bench (Li et al., 2023b), LLaVA-Bench (In-the-Wild) (Liu et al., 2023b) and MM-Vet (Yu et al., 2023). Tab. 2 shows that VLSA achieves obviously better results compared to previous generalist models on MME (+117/+91), SEED (+5.8/+5.6), and MM-Vet (+4.7/+6.3).

Visual-heavy and hallucination evaluations.

In addition, we compare VLSA with LLaVA-Next on document understanding (visual-heavy) tasks AI2D (Kembhavi et al., 2016), ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), and OCRBench (Liu et al., 2024b) in Tab. 4 and Appendix A.1.3. Upon comparing Tab. 4 (6) with (1) and Appendix Tab. 10, we find that despite adopting compressive encoding on visual inputs, VLSA still outperforms the conventional high-resolution approach on tasks requiring fine-grained perceptions. Please also refer to Appendix A.1.2 for evaluations on additional carefully selected benchmarks.

To complement these visual-heavy evaluations, we test V*Bench (Wu & Xie, 2024) and MMVP (Tong et al., 2024b) with a matched LLaVA-1.5 baseline. VLSA improves their scores from 48.68 to 52.77 and from 24.70 to 29.60, respectively (Appendix A.20).

We also examine hallucination directly with Hallusion-Bench (Guan et al., 2024) and AMBER-Discriminative (Wang et al., 2023). The gains of 3.9–7.2 points in Table 6 complement POPE and support reduced hallucination on the evaluated benchmarks.

Cross-architecture generalization and stability.

To demonstrate the generalizability of VLSA, we further integrate VLSA with other MLLM architectures. As shown in Tab. 3, VLSA brings stable improvements to these architectures.

Note that we initialized the models using the officially released weights (without changing the model architectures by integrating our SA-Perceiver) for experiments related to Qwen2.5-VL and Intern2.5-VL, rather than training them from scratch, as we do not have access to their training data. Subsequently, we performed a secondary fine-tuning for them on LLaVA-Next's 790K fine-tuning set. Similarly, SliME is also involved in-house data, thus we reproduce the results using LLaVA-Next's training sets. Each baseline/+VLSA pair shares initialization, training data, secondary fine-tuning, and evaluation. The reported gains are computed within these pairs; official results under their original training settings are provided separately in Appendix A.18.

In Appendix A.8, we assess the integration with existing frameworks LaViT (Jin et al., 2024) and Ovis (Lu et al., 2024) that facilitate modality alignment, and discuss the performance of VLSA across MLLM architectures. We further assess run-to-run stability on three representative benchmarks using seeds 42, 456, and 2024. VLSA achieves 65.37 ± 0.21 on GQA, 83.47 ± 0.15 on VQA^{v2}, and 88.76 ± 0.40 on POPE, compared with reference baseline scores of 64.9, 82.4, and 87.7 (Table 26). The low standard deviations characterize VLSA's stable improvements.

A.1.2 Additional Results on The Effectiveness of VLSA

We further assess the effectiveness of integrating VLSA into existing MLLMs across six carefully curated datasets, in addition to those discussed in Sec. 4. As illustrated in Table 9, we initially focus on benchmarks tailored for high-resolution images, including MME-RealWorld (Zhang et al., 2024c) and Q-bench-A1-single-dev (Wu et al., 2023). The enhancements brought about by VLSA suggest that our method can more effectively leverage the rich visual information available in high-resolution inputs. Additionally, we have selected two benchmarks specifically designed to evaluate hallucinations in MLLMs: Hallusion-Bench (Guan et al., 2024) and AMBER-Discriminative (Wang et al., 2023). The substantial performance improvements achieved with VLSA highlight its capability to efficiently minimize information loss in visual inference and alleviate hallucinations. Furthermore, beyond conventional benchmarks (those involving single-image inputs with textual outputs), we validate the versatility of VLSA by testing it on BLINK-Validation (Fu et al., 2024) and DEMON-Core (Li et al., 2023c). These datasets involve multiple, interleaved, and multimodal instructions, showcasing the required context to complete various tasks. The consistent performance gains confirm that VLSA serves as a general-purpose solution for enhancing visual-semantic understanding.

Table 9: Additional results on six benchmarks to demonstrate the effectiveness of VLSA. High-resolution benchmarks: MME-RealWorld (Zhang et al., 2024c)(MME-Real), Q-bench-A1-single-dev (Wu et al., 2023)(Q-bench); Hallucination evaluation benchmarks: Hallusion-Bench (Guan et al., 2024)(Hall-Bench), AMBER-Discriminative (Wang et al., 2023)(AMBER); Generalizability evaluation benchmarks: BLINK-Validation (Fu et al., 2024)(BLINK), DEMON-Core (Li et al., 2023c)(DEMON). "Avg-Acc" and "Acc" indicate "Average Accuracy" and "Accuracy", respectively.

Method	High-resolution		Hallucination		Generalizability	
	MME-Real (Avg-Acc)	Q-bench (Avg-Acc)	Hall-Bench (Avg-Acc)	AMBER (Acc)	BLINK (Avg-Acc)	DEMON (Avg-Acc)
LLaVA-1.5-HD	26.1	58.6	44.4	75.3	36.9	24.6
+VLSA	30.3	59.1	**49.2**	82.5	38.7	26.6
Δ	**+4.2**	**+0.5**	**+4.8**	**+7.2**	**+1.8**	**+2.0**
LLaVA-Next	29.7	60.2	44.2	78.7	44.6	28.3
+VLSA	**31.8**	**61.7**	48.1	**83.2**	**47.1**	**32.4**
Δ	**+2.1**	**+1.5**	**+3.9**	**+4.5**	**+2.5**	**+4.1**

A.1.3 Performances on Visual-Heavy Benchmarks

To comprehensively evaluate performance on visual-heavy tasks, we further evaluate VLSA on InfoVQA (Mathew et al., 2022) (infographic understanding) and OCRBench (Liu et al., 2024b) (OCR capabilities), in addition to AI2D, ChartVQA, and DocVQA in Tab. 4. These benchmarks

require fine-grained visual perception and demonstrate VLSA's effectiveness on detail-oriented tasks. Even at our standard 336×336 resolution, VLSA surpasses LLaVA-Next with "Any" resolution on both InfoVQA and OCRBench, while reducing computational cost by 77% in FLOPs and 50% in system latency. To further release the potential of VLSA, we further increase the resolution of the downsampled images in compressive encoding to 672×672. The superior performance on OCRBench (563 vs. 510) and InfoVQA (47.2 vs. 44.1) indicates that VLSA can effectively preserve fine-grained visual information necessary for detail-oriented tasks. Please also refer to Tab. 24 for further analysis of the higher-resolution settings.

Table 10: Performances on visual-heavy benchmarks.

Method	Res.	InfoVQA	OCRBench	AI2D	ChartQA	DocVQA
LLaVA-Next	336×336	40.6	419	67.8	44.8	45.4
LLaVA-Next	Any	44.1	510	69.5	67.1	73.7
VLSA	336×336	44.6	554	71.4	67.9	75.2
VLSA	672×672	**47.2**	**563**	**76.3**	**68.5**	**80.1**

A.1.4 Quantitative Analysis of Encoding Information Loss

We use overall reconstruction fidelity and frequency-specific distortion as auxiliary diagnostics of E-IL.

Overall reconstruction quality. We evaluate how faithfully the input image can be reconstructed from the visual embeddings produced by each model using PSNR (Peak Signal-to-Noise Ratio). We randomly select 1,000 images from the training set and train a denoiser for each visual pipeline from the same pretrained initialization under the same training protocol. As shown in Table 11, VLSA improves the average PSNR from 9.02 to 20.18. The higher PSNR shows more faithful reconstruction by the trained encoder–decoder system, providing auxiliary evidence consistent with reduced E-IL. Figure 4 provides corresponding qualitative examples.

Table 11: Reconstruction PSNR for analyzing E-IL. Each visual pipeline uses a separately trained denoiser under the same initialization and training protocol.

Method	**Res.**	**PSNR**
LLaVA-Next	Any	9.02
VLSA	336×336	**20.18**



Figure 4: Qualitative reconstructions from the encoder–decoder systems evaluated in Table 11. VLSA better preserves input-specific content, consistent with lower E-IL.

Frequency-domain analysis. While PSNR summarizes overall reconstruction fidelity, it does not reveal how distortion varies across frequency bands. We therefore analyze 50 randomly selected images and their reconstructions from the same separately trained reconstruction pipelines. We apply a 2D Discrete Cosine Transform (DCT), partition the spectrum into low [0, 1/3), mid [1/3, 2/3), and high [2/3, 1) bands, and compute Spectral Distortion (SD) in each band as

$$\mathrm{SD} = |1 - \mathrm{EPR}|, \tag{6}$$

where EPR (Energy Preservation Ratio) is the ratio of reconstruction band energy to original band energy. Higher SD indicates a larger discrepancy between the original and reconstructed band energy. Because equal band energy can arise from different image content, we interpret SD together with PSNR and the qualitative reconstructions rather than as a standalone information-theoretic measure.

Table 12: Frequency-domain analysis of encoding information loss.

Frequency Band	Range	Baseline SD	VLSA SD	Improvement
Low-frequency	[0, 1/3)	0.583	**0.139**	76.2%
Mid-frequency	[1/3, 2/3)	0.451	**0.343**	23.9%
High-frequency	[2/3, 1)	0.669	**0.234**	64.9%

Key findings:

- The baseline has its largest spectral distortion in the high-frequency band (SD=0.669), associated with fine textures, edges, and text strokes.
- VLSA reduces SD across all frequency bands, including a 64.9% reduction in high-frequency spectral distortion.
- The reduction in high-frequency spectral distortion is consistent with the clearer details in Figure 4 and the stronger visual-heavy results in Table 10, although the contribution of individual frequency bands to downstream gains remains untested.

The higher PSNR, clearer reconstructions, and lower spectral distortion provide auxiliary evidence consistent with reduced E-IL. Because the denoisers are trained separately, these results reflect both visual encoding and reconstruction. Appendix A.17 provides a complementary control by reconstructing unseen images from both representations with a single fixed decoder, supporting lower E-IL under a shared reconstruction model.

A.1.5 Vision-Only Assessment

To further evaluate VLSA's influence on the vision encoder, we assess the visual perception capabilities of the vision encoder using ImageNet-1K (Deng et al., 2009) zero-shot classification in two settings: (1) Open-World Setting, which indicates the label set is not provided, and models are required to generate semantic labels, and (2) Closed-World Setting, which means the candidate class labels are concatenated in prompts. The latter can be applied in encoder-only evaluations based on the similarity between images and class labels.

Table 13: Open-world ImageNet-1K evaluation.

Method	Setting	ImageNet-1K Accuracy
LLaVA-Next	Open-World	31.3
VLSA	Open-World	**36.0**

Table 14: Closed-world ImageNet-1K evaluation.

Method	Configuration	ImageNet-1K Accuracy
LLaVA-Next	Vision encoder only	47.4
VLSA	Vision encoder only	53.2
VLSA	Vision encoder (Frozen LLM)	60.9
CLIP ViT-L/14	Unfinetuned baseline	**74.8**

In the Open-World Setting, VLSA demonstrates 4.7% improvement in zero-shot image classification, indicating enhanced visual understanding capabilities. On the other hand, VLSA's vision encoder achieves 53.2% (+5.8%) and 60.9% (+13.5%) accuracy when trained with a frozen LLM (Cambrian (Tong et al., 2024a) setting) in the Closed-World Setting. While below the unfinetuned CLIP baseline (74.8%), this is expected given the shift in optimization objectives from global contrastive learning to reconstructive detail-focused training.

A.1.6 TEXT-ONLY ASSESSMENT

We also evaluate the LLM backbone (LLaMA3-8B) on three pure-text benchmarks to assess whether visual-language alignment training affects text-only reasoning. As shown in Tab. 15, both the baseline LLaVA-Next and VLSA show performance degradation relative to the original LLaMA3-8B on MMLU (Hendrycks et al., 2021) and HumanEval (Chen et al., 2021), likely due to shifts in the distribution of training data, rather than a flaw in VLSA's design. (We notice that VLSA maintains competitive performance on GSM8K (Cobbe et al., 2021) (mathematical reasoning), where visual information is less relevant.) This degradation can possibly be addressed through incorporating more high-quality text-only data in future training procedures.

Table 15: Text-only task performances.

Method	MMLU (0-shot)	GSM8K (8-shot)	HumanEval (0-shot)
LLaVA-Next	66.2	77.5	60.5
VLSA	65.7	**79.9**	60.9
LLaMA3-8B	**68.4**	79.6	**62.2**

A.2 IMPLEMENTATION DETAILS

As for the LLM backbone, we employ LLaMA3-8B-Instruct (Dubey et al., 2024) for LLaVA-Next, Vicuna-7B-v1.5 (Zheng et al., 2023) for LLaVA-1.5-HD. All of these LLMs feature transformer layers with a dimensionality of $d = 4096$. For the vision encoder, we utilize CLIP-ViT-L/14@336px (Radford et al., 2021), which comprises 24 transformer layers, each with a feature dimension of $d = 1024$. The low-resolution snapshot X_{Lo} referenced in Sec. 3.1 has dimensions of $\mathbb{R}^{3\times336\times336}$, aligning with the input image size required by the vision encoder. The length of the image embedding V, which is input to the LLM, consistently remains at 576, signifying a reduction of up to *five-fold* compared to the original LLaVA-Next and LLaVA-1.5-HD. The epigone described in Equation 4 is implemented as a low-rank MLP with two linear layers and the SiLU activation function. Its input and output dimensions are 4086, while the intermediate dimension is 256. Our development of VLSA builds upon the original LLaVA codebase with minor adjustments. We maintain the two training stages when integrating VLSA with the LLaVA series. In **Stage I**, we prioritize optimizing the SA-perceiver, the Epigone, and the denoising transformer while keeping other parameters fixed. The dataset used in this stage matches the 558K instances previously employed in pre-training LLaVA-1.5-HD and LLaVA-Next. Transitioning to **Stage II**, we jointly optimize these modules, the vision encoder, and the language model using the 665K/790K fine-tuning data employed by LLaVA-1.5-HD/LLaVA-Next. We train all variants of VLSA using 8x NVIDIA A100 80GB GPUs with 500GB of system memory, deployed on the Volcengine cloud platform, completing each process within 19 hours. Table 16 summarizes the training data, trainable modules, and hyperparameters employed during each training stage. We implement greedy decoding with a temperature of 0 during inference to ensure reproducibility.

VLSA is an efficient plug-in that seamlessly integrates into existing MLLMs:

(1) During the training, our cognitive alignment functions as a data augmentation method conducted offline. The additional data labeling processes do not introduce any extra system overhead for training. Moreover, the SA-perceiver in perception alignment can easily replace the projector (such as MLP or Q-former) in other MLLMs, and the efficiency gained through compressive encoding offsets the extra overhead from reconstructive training (Please refer to Fig. 3).

(2) During the inference, all supplementary pretrained models (including VAE, Denoising Transformer, Epigone, and VQ-VAE) are deactivated. Consequently, VLSA does not increase the reasoning overhead.

Table 16: Statistics of training data, trainable modules, and hyper-parameters.

Stage	I	II
Training Data	LLaVA-558K-Pretrain	LLaVA-665K/790k-Finetune
Batch Size	128	64
Warm-up Ratio	0.03	0.03
Weight Decay	0.00	0.00
Trainable Module (learning rate)	perceiver (1e-3) Epigone (1e-3) Denoising Transformer (1e-3)	perceiver (1e-5) Epigone (1e-5) Denoising Transformer (1e-5) Vision Encoder (2e-6) Language Model (1e-5)

Benchmark selection:

(1) To comprehensively compare our VLSA with LLaVA, we first follow LLaVA-1.5's dataset selection and evaluate competing methods on 8 academic-task-oriented datasets (including VQA^{v2} (Goyal et al., 2017), GQA (Hudson & Manning, 2019), VisWiz (Gurari et al., 2018), ST-VQA (Biten et al., 2019), ScienceQA-IMG (Lu et al., 2022), TextVQA (Singh et al., 2019), TextCaps (Sidorov et al., 2020), COCO (Chen et al., 2015)), and 7 instruction-following benchmarks that are tailor-made for MLLMs (including POPE (Li et al., 2023e), MME (Fu et al., 2023), MMBench (Liu et al., 2023c), MMBench-Chinese (Liu et al., 2023c), SEED-Bench (Li et al., 2023b), LLaVA-Bench (In-the-Wild) (Liu et al., 2023b) and MM-Vet (Yu et al., 2023)), in Tab. 1 and Tab. 2.

(2) Besides, we evaluate VLSA and LLaVA on 3 document understanding tasks (including AI2D (Kembhavi et al., 2016), ChartQA (Masry et al., 2022), and DocVQA (Mathew et al., 2021) in Tab. 4) to assess their fine-grained understanding of rich visual information, in Tab. 4.

(3) Additionally, we further evaluate VLSA using benchmarks specifically designed for high-resolution images (MME-RealWorld (Zhang et al., 2024c), Q-bench (Wu et al., 2023)), benchmarks that assess hallucination (Hallusion-Bench (Guan et al., 2024), AMBER (Wang et al., 2023)), and benchmarks that test generalizability (BLINK (Fu et al., 2024), DEMON (Li et al., 2023c)) (Please refer to Appendix A.1).

A.3 More Ablations

This section provides supplementary ablations on (1) the structure of SA-Perceiver, (2) reconstruction models and denoising steps, (3) image encoders, (4) LLM backbones, (5) the relative weights of language-modeling and reconstruction losses, (6) auxiliary supervision tasks, their sampling ratios, and the quantized encoders used to generate targets, (7) snapshot resolution and reconstruction fidelity, (8) additional training stages, and (9) stability across training seeds.

Ablations on the structure of SA-Perceiver. The SA-Perceiver is designed to integrate information from high-resolution images into the features of low-resolution images. Thus, we first employ a cross-attention layer to realize feature interaction. As high-resolution images are divided into multiple sub-images during preprocessing, we subsequently incorporate a self-attention layer after the cross-attention to enhance the modeling of the interrelationships among the aggregated sub-image information in low-resolution images' embeddings. To ensure parameter efficiency, we have omitted certain projection layers typically found in the standard attention mechanism. To demonstrate the effectiveness of our design choices, we conducted the following ablation experiments on the SA-Perceiver in Tab. 17: (ex1) removing the self-attention layer and only retaining the cross-attention layer. (ex2) retaining all linear projections in cross and self-attention, including those for generating keys, queries, values, and outputs.

Our rationale for introducing a learnable parameter q as part of the input of the SA-Perceiver 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. To better substantiate the efficacy of learnable parameter P, we have included an ablation study (ex3) employing global pooling on the outputs of SA-Perceiver to generate the pooled embedding.

Table 17: Ablations on the structure of SA-Perceiver.

Variant	GQA	SQA-I	DocVQA
(ex1) w/o. self-attn	65.1	76.9	72.4
(ex2) full projections	**65.5**	77.3	75.0
(ex3) global pooling	64.9	76.2	73.3
SA-Perceiver	65.3	**77.5**	**75.2**

Table 18: Ablations on different visual projectors.

Variant	GQA	SQA-I	DocVQA
(ex4) MLP	62.1	70.1	73.9
(ex5) Self-Attn	62.3	69.6	74.1
(ex6) Pixel-Unshuffle	64.1	68.2	71.7
(ex7) Q-former	64.5	70.4	62.8
SA-Perceiver	**65.3**	**77.5**	**75.2**

To further illustrate the efficacy of our SA-Perceiver in handling high-resolution representations, we compare its performance against several visual projectors in Tab. 18. Specifically, we evaluate (ex4) an MLP or (ex5) a self-attention that processes high-resolution representations, (ex6) Pixel-Unshuffle (Shi et al., 2016) combined with an MLP, where the pixel-unshuffle operation reduces the visual token count to one-quarter of its original value by resizing the tokens and concatenating them channel-wise, and (ex7) a Q-former that condenses high-resolution representations into a set of learnable queries with the same dimensionality as the low-resolution representations utilized in our SA-Perceiver. We also visualize their influence on language modeling loss during pretraining in Fig. 6.

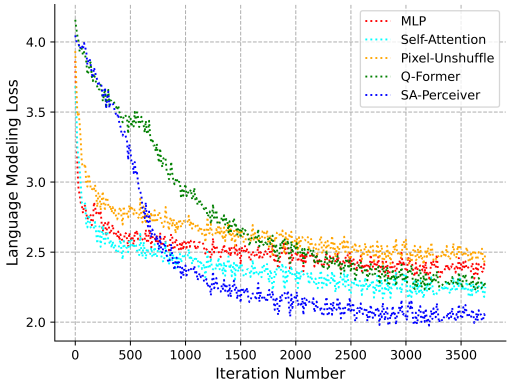

Figure 6: The influence of projectors on language modeling loss.

Ablations on the LDM. To further substantiate the necessity of using text-to-image LDMs, we have conducted new ablations where we implement reconstruction training by replacing the SD with pretrained vision-only models (ex8) VAE decoder (Kingma, 2013), (ex9) VQ-VAE (van den Oord et al., 2017), and (ex10) MAE (He et al., 2022) decoder. The results are reported in Tab. 19. Besides, we validate the performance change when substituting the current LDM with (ex11) stable-diffusion-2-small (Rombach et al., 2022), a version that has fewer parameters, and (ex12) randomly initialized stable-diffusion3-medium trained from scratch. We find that VLSA shows low sensitivity to the scale of LDM, which is important for further adapting VLSA to resource-limited environments. Table 29 further compares LDM- and MAE-based reconstruction, showing that LDM better promotes vision-language alignment despite their comparable reconstruction quality.

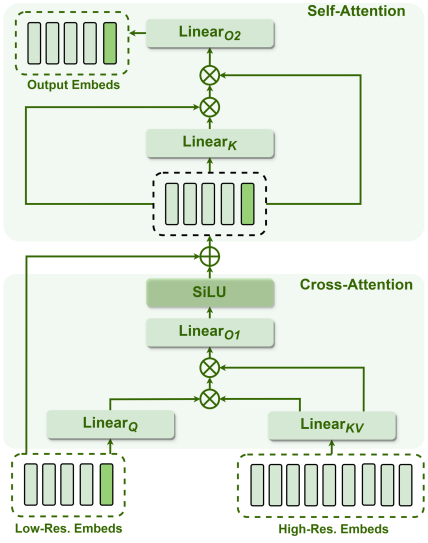

Figure 5: The details of the SA-Perceiver.

Ablations on the image encoder. We evaluate the influence of (ex13) employing the stronger vision encoder SigLIP (Zhai et al., 2023) in Tab. 20. Note that there are discrepancies between the results of (ex14) our reproduced LLaVA-Next(SigLIP) and those reported

Table 19: Ablations on the image generation module.

Variant	GQA	SQA-I	DocVQA
(ex8) VAE	64.7	75.0	74.4
(ex9) VQ-VAE	65.2	74.2	69.3
(ex10) MAE	64.5	74.3	**75.4**
(ex11) stable-diffusion2-base	64.9	77.2	75.1
(ex12) stable-diffusion3-medium (from scratch)	65.0	76.3	74.7
stable-diffusion3-medium	**65.3**	**77.5**	75.2

officially. Due to limitations in computational resources, we have to use a smaller equivalent batch size, which leads to a slight shift in performance.

Table 20: Ablations on the image encoder.

Variant	Vision Encoder	Batchsize	AI2D	ChartQA	DocVQA
LLaVA-Next (Reported)	SigLIP	512	71.6	69.5	78.2
(ex13) LLaVA-Next (Reproduced)	SigLIP	128	71.4	69.2	78.3
(ex14) LLaVA-Next + VLSA	SigLIP	128	**73.7** (+2.3)	**71.2** (+2.0)	**81.1** (+2.9)
LLaVA-Next	CLIP	128	69.5	67.1	75.2
LLaVA-Next + VLSA	CLIP	128	71.4 (+1.9)	67.9 (+0.8)	75.2 (+1.5)

Ablations on the language model. We report the performance of VLSA with the replacement of the language backbone to (ex15) Vicuna1.5-7B, (ex16) Vicuna1.5-13B, and (ex17) Qwen1.5-72B. Additionally, we report the performance of LLaVA-Next with these backbones as references.

Table 21: Ablations on the language model

Variant	LLM	AI2D	ChartQA	DocVQA
LLaVA-Next	Vicuna1.5-7B	66.4	54.7	72.5
(ex15) LLaVA-Next + VLSA	Vicuna1.5-7B	67.5 (+0.9)	55.3 (+0.6)	74.6 (+2.1)
LLaVA-Next	Vicuna1.5-13B	67.0	60.5	75.7
(ex16) LLaVA-Next + VLSA	Vicuna1.5-13B	69.2 (+2.2)	61.9 (+1.4)	79.8 (+4.1)
LLaVA-Next	Qwen1.5-72B	73.4	72.7	79.9
(ex17) LLaVA-Next + VLSA	Qwen1.5-72B	77.1 (+3.7)	77.3 (+4.6)	82.5 (+2.6)

Ablations on the loss ratio. We evaluate the influence of the different ratios between the reconstruction loss $\mathcal{L}_{\text{rec}}$ and the language modeling loss $\mathcal{L}$ in Tab. 22.

Table 22: Ablations on the loss ratio.

Variant	AI2D	ChartQA	DocVQA
(ex18) 0.25: 0.75	71.0	67.2	74.4
(ex19) 0.5: 0.5	70.1	67.5	73.9
(ex20) 0.75: 0.25	68.2	67.6	75.1
1: 1	**71.4**	**67.9**	**75.2**

Ablations on the cognition alignment. To assess the effectiveness and universality of our SSFT objectives in cognitive alignment, we compare them with previous techniques that enhance the LLM's understanding of visual semantics. In particular, we examine (ex21) the additional visual grounding objectives by incorporating the grounded visual chat (GVC) dataset from LLaVA-Grounding (Zhang et al., 2023a) into LLaVA-Next's 790K fine-tuning instances, and (ex22) the additional segmentation objectives achieved by integrating LISA (Lai et al., 2024). All experiments presented in Tab. 23 utilize the same CLIP encoder as employed in our other experiments. Additionally, we exclude perception alignment (which includes compressive encoding and reconstructive training) for VLSA to ensure fair comparisons.

Ablation on Low-Resolution Image Size. To validate VLSA's performance across different efficiency-performance trade-offs, we conduct ablation studies on the resolution of the low-resolution image. The low-resolution image size directly influences computational efficiency while affecting the

Table 23: Ablations on the effectiveness of Cognition Alignment.

Variant	AI2D	ChartQA	DocVQA
(ex21) visual grounding	68.7	67.1	73.0
(ex22) segmentation	68.5	66.8	**74.4**
pixel value+codebook indices	**72.5**	**67.0**	73.9

degree of Original Information Loss (O-IL). Results show that increasing low-resolution dimensions impressively improves VLSA's performance on visual understanding tasks.

Table 24: Resolution ablation on low-resolution image size. The "Res." column indicates the maximum size of the low-res image input to the MLLM.

Method	Res.	GQA	SQA-I	DocVQA
LLaVA-Next	Any	64.9	74.6	73.7
VLSA (current)	336×336	65.3	77.5	75.2
VLSA	336×672	66.8	78.1	77.8
VLSA	336×1008	67.9	80.4	79.2
VLSA	672×672	**69.2**	**80.8**	**80.1**

These results demonstrate that VLSA's design principle of maintaining computational efficiency can be flexibly balanced with performance improvements by adjusting the low-resolution image size. When efficiency constraints are relaxed, VLSA achieves substantially greater improvements over the baseline.

Ablation on Training Stages. To validate the contribution of auxiliary warmup stages, we compare the standard two-stage training used in baseline LLaVA with a three-stage variant that includes explicit warmup of the Denoising Transformer component. The auxiliary warmup stage is enabled only for VLSA:

Table 25: Training stage ablation. The three-stage variant includes an explicit warmup stage for the Denoising Transformer before the main visual instruction tuning.

Method	Res.	Stages	GQA	SQA-I	DocVQA
LLaVA-Next	Any	2	64.9	74.6	73.7
VLSA (current)	336×336	2	65.3	77.5	75.2
VLSA	336×336	3	65.6	78.2	75.6
VLSA	672×672	2	69.2	80.8	80.1
VLSA	672×672	3	**69.5**	**81.3**	**80.2**

This ablation demonstrates that VLSA achieves strong results with standard two-stage training, while a specialized warmup stage can provide modest additional improvements for practitioners seeking maximum performance. (The main text uses two-stage training for fair comparison with baseline LLaVA.)

Stability Across Training Seeds. To validate the reliability of improvements, we report mean and standard deviation across three runs with different random seeds (42, 456, 2024) on three representative benchmarks.

Table 26: VLSA performance across three training seeds, reported as mean ± standard deviation. The baseline and 672 × 672 rows are single-run reference results.

Method	**Res.**	**GQA**	**VQA-V2**	**POPE**
LLaVA-Next	336×336	62.3	77.1	85.4
LLaVA-Next	Any	64.9	82.4	87.7
VLSA	336×336	65.37 ± 0.21	83.47 ± 0.15	88.76 ± 0.40
VLSA	672×672	69.2	85.1	89.5

The standard deviations of 0.21, 0.15, and 0.40 indicate low variation across the three VLSA runs, whose means exceed the corresponding reference baseline scores. Since the baseline is reported

without seed variance, this comparison characterizes VLSA's stability rather than the statistical significance of paired training-run differences.

SSFT Objective Ratio Ablation We ablate the ratio between codebook index prediction and pixel value prediction in SSFT objectives. The ratio indicates how frequently each auxiliary task is inserted into the training data (e.g., 10:1 means for every 10 samples with codebook prediction, 1 sample has pixel prediction).

Table 27: Ablation on SSFT objective ratio (codebook : pixel).

SSFT Ratio	AI2D	SQA	ChartQA
1:2	65.5	67.9	65.8
1:1	68.7	71.8	66.0
2:1	69.9	74.3	65.8
5:1	71.1	77.2	68.2
10:1	**71.4**	**78.6**	**67.9**

Conclusion: The 10:1 setting gives the strongest AI2D/SQA results among the tested ratios, while 5:1 is slightly better on ChartQA. We therefore adopt 10:1 as a practical balance across these tasks, using pixel prediction less frequently than codebook prediction.

LDM Denoising Steps Ablation We ablate the number of denoising steps used in LDM-based reconstructive training to evaluate the trade-off between training cost and downstream performance.

Table 28: Ablation on LDM denoising steps.

Denoising Steps	Training Time	GQA	SQA-I	POPE
1 step	18.9h	65.3	77.5	88.6
2 steps	22.1h (+17%)	65.5	76.9	88.7
5 steps	28.4h (+50%)	65.5	77.2	89.1

Conclusion: Multi-step denoising yields negligible downstream performance improvements while substantially increasing training time. Single-step denoising provides effective training signals for the vision encoder without the computational overhead of iterative refinement. This aligns with the principle that reconstructive training aims to provide gradient signals rather than generate high-quality images.

Comparison of Reconstruction Models: LDM vs. MAE To understand why LDM-based reconstruction outperforms simpler alternatives, we compare the vision-language semantic alignment (with CKA, Eq. 12) achieved by LDM-based and MAE-based reconstructive training.

Table 29: Vision-language alignment comparison: LDM vs. MAE (CKA).

LLM Layer	MAE-based	LDM-based	Improvement
Layer 4	0.26	**0.30**	+0.04
Layer 8	0.38	**0.42**	+0.04
Layer 12	0.53	**0.57**	+0.04
Layer 16	0.56	0.55	-0.01
Layer 24	0.84	**0.88**	+0.04
Layer 32	0.81	**0.85**	+0.04

Note: Although MAE-based and LDM-based training achieve comparable reconstruction quality (PSNR: 20.76 vs. 20.18), LDM-based training yields higher CKA in five of the six tested layers. These results show that LDM more effectively promotes vision-language alignment despite providing similar reconstruction fidelity.

Conclusion: This result supports our motivation for using a pretrained text-to-image LDM rather than a vision-only reconstruction model. Because the LDM has learned associations between visual content and language during pretraining, conditioning its reconstruction on visual embeddings encourages

those embeddings to preserve image information in a form that is more compatible with language representations. This provides a plausible explanation for the stronger downstream performance of LDM-based reconstructive training. Appendix A.19 provides complementary evidence by comparing pretrained and randomly initialized SD3.

Sensitivity to VQ-VAE Model Choice To evaluate VLSA's robustness to the choice of quantized encoder, we compare three different models for generating codebook indices in SSFT objectives.

Table 30: Ablation on quantized encoder choice.

Quantized Encoder	AI2D	SQA	ChartQA
VQ-VAE	71.4	78.6	67.9
VQ-GAN	71.7	78.3	**68.0**
RQ-VAE	**72.1**	**78.8**	67.5

Conclusion: Performance varies by at most 0.7 points across the three tested quantized encoders, indicating robustness to the choice among these label generators.

Resolution Ablation with PSNR Analysis To quantify the relationship between input resolution, information preservation, and downstream performance, we measure reconstruction PSNR at different input resolutions.

Table 31: Input resolution impact on information preservation and performance.

Input Resolution	Reconstruct PSNR	GQA	Interpretation
168×168	11.26	43.8	High O-IL
336×336	19.80	62.4	Moderate O-IL
672×672	20.27	65.2	Low O-IL

Conclusion: Increasing the input resolution consistently improves both reconstruction PSNR and GQA performance. This joint trend supports the role of O-IL: aggressive downsampling discards visual information before encoding, whereas a higher-resolution input leaves more information recoverable and available for downstream reasoning. Because reconstruction PSNR is measured after visual encoding and GQA after the full inference pipeline, this experiment supports O-IL as a performance bottleneck without isolating its effect from subsequent E-IL and D-IL.

A.4 Design Rationale and Technical Details

VLSA is designed to preserve visual information during encoding and make it usable for response generation (Fig. 2). **Perception Alignment** consists of compressive encoding and LDM-based reconstructive training. The former gathers high-resolution context into compact visual embeddings, while the latter encourages these embeddings to retain input details that compression might otherwise discard and promotes their compatibility with language. Preserving information alone, however, does not ensure that the LLM can interpret it. **Cognition Alignment** therefore trains the LLM to predict codebook indices and pixel values alongside instruction responses, strengthening its understanding of high- and low-level visual semantics. Table 32 summarizes the components and objectives.

A.4.1 Design Rationale for SA-Perceiver Architecture

The SA-Perceiver employs shared weight matrices for Keys and Values in the cross-attention component. This design choice prioritizes parameter efficiency while maintaining performance. Specifically:

$$K_{\mathrm{c}} = \bar{V}_{\mathrm{Hi}} w_{\mathrm{kv}}, \quad V_{\mathrm{c}} = \bar{V}_{\mathrm{Hi}} w_{\mathrm{kv}} \tag{7}$$

Here tokens are stored as rows, and the key and value streams share the projection w_{kv}. In the self-attention component, we omit the standard Query and Value projection matrices to further reduce computational overhead. While non-standard, these simplified attention mechanisms adhere to the fundamental attention computation. The effectiveness of this design arises from the observation that

Table 32: **Components and objectives of VLSA.** The loss categories distinguish where information is lost, without assuming statistical independence or additivity. Each component may affect more than one category.

Alignment	Component / objective	Role
Perception Alignment (O-IL, E-IL)	Compressive encoding (SA-Perceiver)	Aggregate high-resolution context into a short visual sequence.
	LDM-based reconstructive training	Preserve input content through reconstruction from visual embeddings.
Cognition Alignment (D-IL)	Codebook-index prediction (SSFT)	Predict VQ-VAE indices to learn high-level visual semantics.
	Pixel-value prediction (SSFT)	Predict RGB values to learn low-level visual semantics.

the SA-Perceiver, as a lightweight adapter module, does not require the strong feature transformation capabilities of full-rank projection matrices. This finding is supported by ablation studies in Tab. 17 (ex2), which compare the simplified design with variants using standard independent Q, K, and V matrices.

As for the output global embedding P, this embedding is obtained by aggregating information from both high-resolution and low-resolution embeddings through the SA-Perceiver, using a learnable query vector q as the aggregation token. This process can be analogous to extracting the CLS token in CLIP. The requirement for generating P stems from the LDM, Stable Diffusion 3 applied in reconstructive training, which necessitates a pooled embedding to capture global semantic information for text-to-image diffusion conditioning. Through dedicated ablations (Tab. 17 ex3), we validated that the learnable query approach outperforms simple global average pooling for this purpose, as it learns task-specific aggregation patterns during end-to-end training.

A.4.2 Analysis of Latent Diffusion Model Choice

The reconstructive training uses pretrained Stable Diffusion 3-medium rather than simpler autoencoder models. The key motivation is that text-to-image LDMs are optimized to map visual features to linguistic semantic space. During training, we require the LDM to reconstruct images from visual embeddings, thus naturally aligning these embeddings with concepts representable in natural language. This linguistic semantic alignment proves particularly beneficial for subsequent language modeling. Tab. 19 compares LDM-based reconstruction with alternative approaches. It demonstrates that while alternative models can also reduce E-IL, pretrained LDM represents the optimal choice.

As for the concern involving the computational cost of LDM-based reconstruction, we address this from three perspectives: (1) Single Denoising Step: Unlike image generation requiring multiple diffusion steps, reconstructive training performs only a single denoising iteration per training batch. (2) Training-Only Component: The LDM participates exclusively in the training process, bringing zero overhead during inference. (3) The computational savings from our compressive encoding substantially exceed the single-step LDM overhead.

A.4.3 Comparison with Q-Former and Mini-Gemini

Relationship to Q-Former/Perceiver. SA-Perceiver is a Q-Former/Perceiver-style resampler. Both compress visual features into fewer tokens, and end-to-end training alone cannot guarantee lossless encoding. In the early stages of training, the randomly initialized fixed-length query vectors in Q-former cannot effectively filter redundancy in visual features and retain key information, thus introducing more severe E-IL compared to MLP-based methods (which do not perform resampling, only perform mapping).

To reduce the risk of discarding visual details during resampling, SA-Perceiver forms its queries from spatial low-resolution CLIP embeddings and a learnable global query, rather than relying solely on randomly initialized learnable query vectors. The comparisons in Fig. 6 and Tab. 18 (ex7) show that the Q-former converges more slowly during training and achieves significantly weaker performance compared to other projectors on the perception-oriented benchmark DocVQA. (Note that Q-former's superior performance on some global understanding tasks only indicates better global information

aggregation capabilities and does not contradict the conclusion that it produces more severe detail information loss during encoding.)

Comparison with Mini-Gemini. Mini-Gemini's patch information mining (Li et al., 2024b) and SA-Perceiver both use low-resolution features as queries for high-resolution information while controlling visual tokens. Their key differences are as follows.

(1) **Visual encoders.** Mini-Gemini uses a frozen CLIP-ViT encoder for low-resolution inputs and a separate frozen ConvNeXt encoder for high-resolution inputs. VLSA uses the same CLIP encoder for the snapshot and patches, and its training objectives update the visual encoding and alignment modules.

(2) **High/low-resolution interaction.** In Mini-Gemini, each query mines its corresponding local high-resolution sub-region. In SA-Perceiver, low-resolution tokens together with a global query cross-attend to spatially ordered, aggregated high-resolution patch tokens, followed by self-attention.

(3) **Training objective.** Mini-Gemini optimizes its patch information mining projector through VLM instruction tuning. In VLSA, the projector outputs also condition a pretrained LDM, with $\mathcal{L}_{\text{rec}}$ constraining the input information retained in these embeddings.

The shared use of low-resolution queries to retrieve high-resolution information is therefore not itself our contribution. If a backbone already compresses high-resolution features into a compact visual sequence, VLSA can retain its existing compression module instead of adding SA-Perceiver, apply reconstructive training to the resulting embeddings, and incorporate Cognition Alignment during instruction tuning.

A.4.4 VQ-VAE in Cognition Alignment

The VQ-VAE model used in Cognition Alignment employs a codebook of size **4096**. This codebook size balances semantic breadth with computational tractability. However, VLSA does not depend exclusively on VQ-VAE. Our Cognition Alignment represents a structural advantage inherent to the task definition, rather than an improvement resulting from the use of particular powerful models. Any quantization encoder that can provide discrete labels can be applied to Cognition Alignment.

A.5 Analysis of Efficiency Evaluation

In Fig. 3, we compare the latency and Flops of the baseline model and VLSA during the feedforward process of processing 1,000 images across various input resolutions. Despite adding multiscale visual feature interactions in SA-Perceiver and integrating reconstructive training, our VLSA still showcases impressive efficiency. Below, we outline two potential reasons that may contribute to this phenomenon.

(1) As for 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 five-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 25. (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.)

(2) During our reconstructive training, 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 encoding.

A.6 More Evaluations on Modality Alignment

We quantitatively evaluate the efficiency of VLSA in facilitating modality alignment. Specifically, we trained three variants (LLaVA-Next, LLaVA-Next with Q-former as the projector, and LLaVA-Next integrated with our VLSA) using 790k visual instruction tuning data, saving a checkpoint every

Table 33: Language modeling losses on the image captioning task based on codebook indices.

Epoch	0.2	0.4	0.6	0.8	1.0
(ex23) LLaVA-Next + VLSA (w/o SSFT)	1.94	1.29	1.12	0.93	0.88
LLaVA-Next + VLSA	0.73	0.62	0.57	0.52	0.54

2435 iterations (one epoch consisting of 12,176 iterations). We then assessed the quality of modality alignment using the following two metrics, with higher values indicating better alignment:

(1) CLIP Score: We randomly selected 1,000 images for each checkpoint and prompted the model to generate detailed captions, then employed the CLIP model to evaluate the correlation logits between the input images and their corresponding generated captions. The final score was determined by calculating the average of these logits. Results are shown in Fig. 7 (Left).

(2) Benchmark Performance: For each checkpoint, we test their performance on the GQA benchmark. Results are shown in Fig. 7 (Right).

These evaluations enable us to validate the impact of our proposed method on modality alignment by analyzing the variations in model performance throughout the training process. The results demonstrate that VLSA achieves modality alignment more quickly and effectively.

A.7 The Effectiveness of Predicting Codebook Indices

We explore the rationale underpinning the effectiveness of predicting codebook indices. Pertaining to the assertion articulated in Sec 3.2, the mapping between natural language and codebook indices may originate from the multi-task learning and transfer learning capabilities inherent in LLMs. This suggests that LLMs can utilize visual semantics as intermediate representations, thereby facilitating the connection between natural language and codebook indices. To substantiate this claim, we propose additional experiments: specifically, we fine-tune the well-trained VLSA (integrated into LLaVA-Next), alongside its variant that omits cognition alignment, on a task that necessitates generating captions based on codebook indices rather than direct image inputs. As shown in Tab. 33, the comparison reveals that VLSA demonstrates a significantly lower loss relative to its variant, suggesting that the prediction of codebook indices indeed enhances the mapping between natural language and Codebook indices. Meanwhile, codebook indices prediction aids LLMs in comprehending the relationship between visual semantics and codebook indices, ultimately bolstering LLMs' capacity to correlate visual semantics with natural language.

A.8 More Model Integration Studies

A.8.1 Integration with Previous Alignment Methods: LaViT

LaViT (Jin et al., 2024) adopts a different approach to visual-language alignment, which reconstructs frozen vision encoder outputs rather than raw images like our VLSA. While its structural constraints

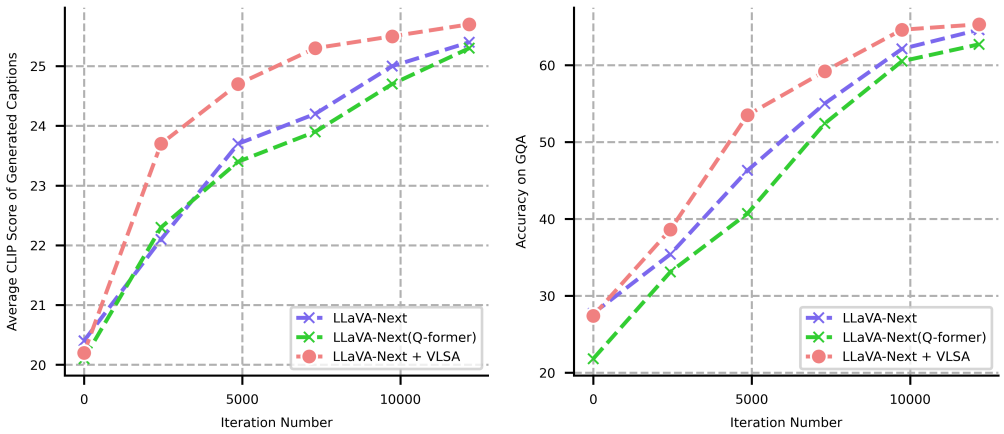

Figure 7: **(Left)** Changes in average CLIP Score during training. **(Right)** Changes in accuracy on GQA benchmark during training.

prevent full VLSA integration, we evaluate the compatibility of our Cognition Alignment objectives with LaViT on GQA, VQA-v2, VizWiz, and NoCaps (Agrawal et al., 2019):

Table 34: LaViT integration with Cognition Alignment. Note that results are preliminary; further hyperparameter tuning specific to LaViT could enhance performance gains.

Method	GQA	VQA-V2	VizWiz	Nocaps
LaViT	46.8	66.0	38.5	114.2
LaViT + Cognition Alignment	46.8	**66.2**	**42.3**	**118.7**

The compatibility demonstrated here indicates that VLSA's Cognition Alignment objectives (SSFT) represent a generalizable contribution orthogonal to specific visual encoding strategies, supporting broader applicability across diverse MLLM architectures.

A.8.2 Integration with Previous Alignment Methods: Ovis

Ovis (Lu et al., 2024) employs a learnable visual vocabulary to discretize image tokens, strengthening formal consistency between vision and language modalities. This approach is orthogonal to VLSA's design philosophy. We demonstrate integration of VLSA's Perception and Cognition Alignment components with Ovis on MME, MMBench, MMMU (Yue et al., 2024), and HallusionBench:

Table 35: Ovis integration with VLSA. Since Ovis replaces traditional adapters with a visual vocabulary, we omitted SA-Perceiver and used Ovis's discretized visual embeddings as reconstruction cues.

Method	MME	MMB-EN	MMMU-Val	HallusionBench
Ovis-Llama3-8B	2009	77.4	44.7	61.1
Ovis + VLSA	**2036**	77.4	**45.2**	**64.2**

These results demonstrate that VLSA's core contributions (reconstructive training via Perception Alignment and SSFT via Cognition Alignment) provide complementary benefits to architectures already employing visual quantization, confirming the generalizability of our approach.

A.8.3 Performance Variation Across MLLM Architectures

VLSA demonstrates varying performance gains across different MLLM architectures due to multiple factors:

(1) The hyperparameters used in the paper were specifically tuned for LLaVA-Next. When these settings are applied to alternative architectures, suboptimal configurations can result.

(2) Different MLLMs train on sets that vary in both the quantity and diversity of images. However, our Cognition Alignment requires the model to learn to predict the codebook indices of images, so the benefits of this objective will depend on the scale and distribution of the images used.

(3) VLSA enhances model performance by minimizing information loss. Different models experience varying degrees of information loss, which affects the magnitude of the gains achieved through VLSA. Nevertheless, the current experiments demonstrate that VLSA consistently yields positive gains across various MLLM architectures, underscoring the effectiveness and value of this approach.

Despite these variations, VLSA consistently provides positive improvements across architectures (LLaVA-Next, MiniGemini, SliME, Qwen2.5-VL, InternVL2.5, LaViT, Ovis), demonstrating robust effectiveness and practical value as a general-purpose alignment enhancement plugin.

A.9 Language Modeling Loss

For each instruction in the augmented dataset, we encode it into word token embeddings I_{emb}, which are then concatenated with image embeddings V, serving as inputs for the LLM to generate a response R, which is formulated as:

$$R = g_{\text{LLM}}\left(\,[V,\ I_{emb}]\,\right), \tag{8}$$

where g_{LLM} symbolizes the LLM's reasoning process. The optimization of language modeling can be formulated as:

$$\operatorname{argmin} \mathcal{L}\left(R, Target;\, \theta_{\mathrm{LLM}}\right). \tag{9}$$

Here, $Target$ refers to the augmented ground-truth answer, $\mathcal{L}(\cdot)$ denotes the language modeling loss, and θ_{LLM} represents the LLM's parameters. $\mathcal{L}(\cdot)$ is defined as:

$$\mathcal{L} = -\sum_{i=1}^{B}\sum_{j=1}^{K+1} \log p\left(y_j^i | V^i,\, I_{emb}^i,\, Target_{0:j-1}^i;\, \theta_{\mathrm{LLM}}\right), \tag{10}$$

where B denotes the batch size, and K is the length of the response R. The final loss for the entire system is determined through the summation of $\mathcal{L}$ and $\mathcal{L}_{\mathrm{rec}}$.

A.10 Comparison with Native Multimodal Frameworks

To gain a deeper understanding of VLSA, we can compare it with native multimodal models such as Chameleon (Team, 2024), Show-o (Xie et al., 2024), and Janus-Pro (Chen et al., 2025a):

VLSA operates as a multimodal model centered around the LLM, enabling the LLM to comprehend visual inputs through a limited set of multimodal data. The core of VLSA lies in its efficient pursuit of modality alignment between vision and language. In contrast, native multimodal models like Chameleon necessitate a fundamental update to their LLM-parameter-initialized multimodal decoders using extensive multimodal datasets, although they possess the capability to perform multimodal understanding and generation tasks concurrently.

From an architectural perspective, the LLM decoder in VLSA processes both textual and visual features that have been refined through projectors, including MLP, Q-Former, and our SA-Perceiver. In contrast, multimodal decoders in models such as Chameleon employ unaligned visual and textual features. This difference likely accounts for the higher data demands of the latter, as they must adapt their decoders to a shifted distribution, despite generally having greater potential.

Regarding training objectives, VLSA aims to 1) minimize information loss incurred by the visual encoder and projectors through reconstructive training and 2) enhance the LLM's understanding of visual semantics via an unimodal autoregressive task, which predicts the textualized discrete visual semantic labels (codebook indices) of images. Collectively, these objectives optimize modality alignment. On the other hand, models like Chameleon enable decoders to directly process visual features through tasks involving image reconstruction (generation) or multimodal autoregression.

A.11 Comparison with Early fusion Methods

Regarding QA-ViT (Ganz et al., 2024), EMMA (Ghazanfari et al., 2025), BRAVE (Kar et al., 2024), and mPLUG-Owl2 (Ye et al., 2023) mentioned in Sec 2: These works adopt a similar approach to enhancing modality alignment by modifying the location of Visual-Language (VL) feature fusion. Conceptually, methods like LLaVA, Qwen-VL, and InternVL series, which fuse VL features only within the LLM, are late-fusion. The four mentioned works allow preliminary VL feature interaction before input to the LLM, termed early fusion:

(1) QA-ViT: Implements VL early fusion by adding cross-modal attention layers within the visual encoder.

(2) EMMA: Uses the "modality adaptation" module based on linear projections for VL early fusion.

(3) BRAVE: Fuses features from multiple visual encoders with text features in an improved Q-Former (MEQ-Former) for VL early fusion.

(4) mPLUG-Owl2: Employs a similar idea to EMMA, but first uses a compression module to reduce visual sequence length and employs a more complex cross-attention-based modality adaptation module.

Compared to late fusion, early fusion better filters visual information to focus on user requests, boosting performance on standard VQA tasks. However, this approach is a double-edged sword: performance degrades when user queries are ambiguous/overly verbose (e.g., SQA) or during fine-grained perception tasks (e.g., MME, ChartQA, InfoVQA). This likely explains why top-performing

MLLMs like Qwen2.5-VL (Bai et al., 2025) and InternVL-2.5 (Chen et al., 2025b) avoid early fusion. The table below reports replication results for QA-ViT, showing significant performance drops versus baseline on SQA-I (containing verbose lectures hinting at solutions) and MME & InfoVQA (requiring fine-grained perception).

Table 36: Comparison of the early fusion method with the baseline.

Method	VQA^{v2}	GQA	MM-VeT	InfoVQA	COCO-cap	SQA-I	MME
(ex24) LLaVA-1.5+QA-ViT	**77.8**	**62.9**	**31.6**	25.5	109.3	66.6	1394
LLaVA-1.5	76.6	62.0	30.6	**26.1**	**109.4**	**69.8**	**1463**

In contrast, our VLSA improves modality alignment from a different perspective: explicitly constraining information loss during the reasoning. (Early fusion methods can be seen as reducing the harm of lost information based on text-visual relevance, but cannot directly minimize the information loss.) As demonstrated in the paper, VLSA delivers consistent performance gains across diverse downstream tasks without harming specific capabilities, making it more generalizable. Crucially, VLSA seamlessly integrates with existing MLLM architectures. (We have validated VLSA in LLaVA-Next, MiniGemini, SliME, Qwen2.5-VL, and InternVL2.5.) Particularly for dynamic high-resolution methods that already employ visual feature compression (e.g., Qwen-VL, InternVL2.5), VLSA can omit SA-Perceiver, integrate Reconstructive Training and Cognition Alignment into their training phase to enhance performance further. Critically, during deployment, this integration introduces no structural changes and zero computational overhead to the base model.

A.12 Comparison with existing Reconstructive Training Objectives.

The key distinction between concurrent work Ross (Wang et al., 2024a) and VLSA is the source of reconstruction cues: Ross uses image tokens processed by the LLM, while VLSA uses the output of the alignment projector. (We also initialize the denoising module with a pre-trained text-to-image LDM vs. Ross's random initialization.) Indeed, the Ross-style structure was considered as one of the alternatives during the development of VLSA. Although using the LLM's output as the reconstruction cue can also help mitigate information loss, our experiments demonstrated that its performance was inferior to the solution we adopted:

Table 37: Comparison of different clues for reconstruction.

Variant	GQA	SQA-I	DocVQA
(ex25) LLM's outputs as clues (Ross-style)	62.9	75.1	75.0
projector's outputs as clues (ours)	65.3	77.5	75.2

Two plausible reasons can be identified for this performance discrepancy. First, as discussed in Sec 3.1, using the projector's outputs as reconstruction cues leverages the pre-trained knowledge of the text-to-image LDM, enhancing the semantic alignment between visual and textual tokens before input into the LLM. The stronger this alignment, the more effectively the LLM can process the visual information that visual tokens contain. Second, the Ross-style structure necessitates the LLM learning to convey visual information. When optimized alongside our cognition alignment objectives, this learning objective may encounter challenges related to multi-objective optimization. Additionally, our further research on VLSA reveals that the Ross structure causes noticeably more severe forgetting on text-only tasks after visual instruction tuning (since this issue lies outside the scope of this paper, we will not discuss it in detail here).

A.13 Cognition Alignment Templates

We report lists of instructions used to predict high-level and low-level visual semantics in Tab. 39 and Tab. 38. They present the same meaning with natural language variance. For each input image-conversations pair in the 790K visual instruction tuning dataset from LLaVA-Next (Liu et al., 2024a), we randomly sample a template from each list and expand it with information on the current image's dimensions and the region we care about. For templates predicting high-level visual semantics, we also provide a hint regarding the length of the targeted VQ indices based on the specific VQ-VAE used. The expanded templates are then combined with $Target_{\mathrm{VQ}}$ and $Target_{\mathrm{PX}}$ in Sec. 3.2 to create

Table 38: The list of instructions for predicting pixel RGB values of images.

- For each pixel in this image, please provide the corresponding RGB value.
- I would like to know the exact RGB value of every pixel depicted in this image.
- Could you please list the RGB values for all pixels in this image?
- Please give me a detailed breakdown of the RGB values for each pixel in this image.
- I need the RGB value of each and every pixel in this image, can you help?
- Would it be possible to obtain the RGB values for all the pixels in this image?
- Please provide a comprehensive report on the RGB values of each pixel in this image.
- I'm looking for the RGB value of each pixel in this image, could you assist?
- Can you please specify the RGB value for every single pixel in this image?
- I require the RGB values of all pixels in this image, please provide them.
- Please tell me the RGB value of each pixel in this image, from top left to bottom right.
- I'm interested in the RGB values of each pixel, can you give me that information for this image?
- Could you please share the RGB values for each pixel in this image with me?
- I would appreciate it if you could tell me the RGB value of each pixel in this image.
- Please provide the RGB values for all pixels in this image, in a readable format.
- I'm seeking the RGB value of each pixel in this image, can you help me out?
- Would you be able to give me the RGB values for each pixel in this image?
- I need to know the RGB value of every pixel in this image, please provide the details.
- Please tell me the RGB value of each and every pixel that makes up this image.
- I'm looking to obtain the RGB values for all pixels in this image, can you assist me?

Table 39: The list of instructions for generating VQ indices of images.

- Provide the VQ indices that would represent the essential features of this image, capturing its key visual elements.
- Derive the VQ indices that capture the diversity and complexity of this image's visual information.
- Describe the VQ indices that effectively represent the visual essence of this image.
- Assuming a VQ-based representation, what are the indices that would summarize the key aspects of this image?
- What are the VQ indices that would allow for the reconstruction or recognition of this image, based on its most salient features?
- Imagine the VQ indices as a codebook of visual elements, and devise the sequence that best represents the narrative or story within this image.
- What are the VQ indices that summarize the most prominent shapes, lines, and forms found within this image, forming a visual abstract?
- Formulate the VQ indices as a set of vectors that represent the hierarchical organization of visual information in this image, from general to specific.
- Assuming a VQ-VAE model, what would the latent space representations (VQ indices) of this image look like, encoding its generative factors?
- Describe the VQ indices that would enable a machine learning model to recognize or classify this image based on its visual characteristics and patterns.

additional conversations, which are inserted at random positions within the original conversations. We report one example used in visual instruction tuning with cognition alignment in Tab. 40.

A.14 More Case Studies

We provide more comparisons between LLaVA-Next before and after integrating our VLSA on visual writing tasks in Fig. 9 and Fig. 10, and on visual question answering tasks in Fig. 8. These qualitative results substantiate the effectiveness of VLSA in reducing information loss.

Table 40: One example used in visual instruction tuning with cognition alignment.

Conversations:

• from: **Human**

<image> What are the colors of the bus? Answer the question with gpt-T-COCO format.

• from: **GPT**

The bus in the image is white and red.

==========

• from: **Human**

What feature can be seen on the back of the bus?

• from: **GPT**

The back of the bus features an advertisement.

==========

• from: **Human**

Imagine a hypothetical encoding process for the upper left quarter of this image. What would the resulting VQ indices look like, summarizing its content? Note that the image resolution is [3, 672, 672], so you should generate 1764 indices

• from: **GPT**

<vq_start>3492, 3492, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, 532, ..., 2054, 3764, 4015, 570, 2675, 344, 3369, 580, 2413, 282, 1653, 2826, 1593, 2253, 1988, 3105, 3488, 1455, 1945, 3069, 1118, 1130, 1130, 2571, 2883, 3160, 1433, 3428, 815, 901]<vq_end>

==========

• from: **Human**

Is the bus driving down the street or pulled off to the side?

• from: **GPT**

The bus is driving down the street, which is crowded with people and other vehicles.

==========

• from: **Human**

Please tell me the RGB value of each and every pixel that makes up the upper left quarter of this image. Note that the image resolution is [3, 672, 672].

• from: **GPT**

<rgb_start>[[205, 203, 191], [203, 201, 189], [202, 200, 188], [183, 183, 171], [208, 209, 195], [186, 183, 166], ..., [205, 203, 191], [204, 202, 190], [203, 201, 189]]<rgb_end>

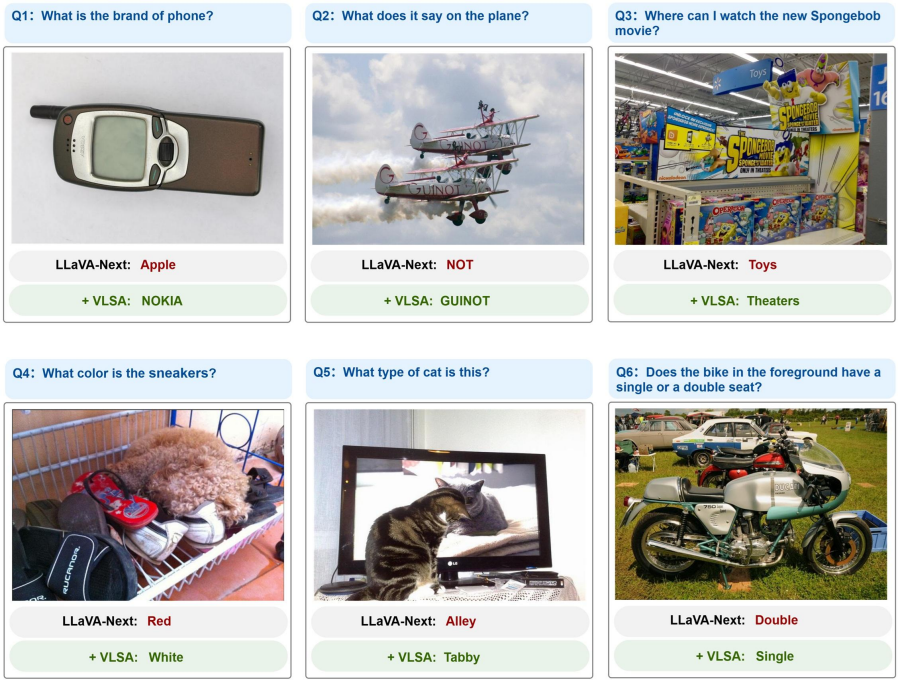

Figure 8: VLSA reduces the loss of visual information and mitigates hallucinations in VQA tasks.

User:

Examine the image closely and identify as many details as you can. Pay attention to landmarks, architectural features, natural elements, signs, and any other clues that might help determine the location. Based on these details, tell me where this place is.



LLaVA-Next

The image shows a large, multi-story building with a traditional architectural style, featuring multiple levels and a flat roof. The building is illuminated, suggesting it is a significant structure, possibly a landmark or a place of interest. It is situated on the edge of a body of water, which reflects the building's lights and the surrounding landscape. The reflection is visible on the water's surface, indicating that the water is calm. In the background, there are snow-capped mountains, suggesting a mountainous region. The sky is dark, which could indicate either dusk or dawn, or it could be a cloudy day. There are no visible texts or inscriptions in the image. The style of the image is a photograph with a focus on the building and its surroundings."

LLaVA-Next + VLSA

The image depicts the **Potala Palace**, a renowned landmark located in Lhasa, Tibet. The palace is a large, multi-tiered structure with a distinctive red and white color scheme. The building has a series of rectangular sections with flat roofs, and the facade appears to be made of **stone or concrete**. The structure is situated on a hillside, and there are **trees in the foreground**, reflecting on the water below. It is surrounded by a serene lake and snow-capped mountains in the background. The palace is illuminated at night, casting a **warm glow** that contrasts with the cool blue of the sky. The image is a photograph, and there are no visible texts or distinctive brands that provide additional information about the location. The Potala Palace is a significant cultural and historical site, known for its grandeur and architectural beauty.

Figure 9: VLSA provides more accurate object recognition and detailed descriptions.

User:

Compose an engaging story about this picture, explaining where this place is and highlighting the animals and their activity.



LLaVA-Next

In the vast expanse of the African savanna, a young **leopard** is on the hunt. The leopard, with its agile body and sharp senses, is crouched low to the ground, stalking its prey. It is in pursuit of a group of three antelopes.

The presence of the leopard indicates that this area is rich in prey, making it a suitable habitat for predators like the leopard. The animals' activity of running and chasing each other showcases the natural behavior and interactions between these species in their shared environment.

LLaVA-Next + VLSA

In the vast expanse of the African savanna, a group of three antelopes are captured in a moment of pure exhilaration. They are running across **a grassy field**, their **hooves pounding the earth** as they are chased by a **cheetah**. The cheetah, with its distinctive **black and yellow spots**, is in the foreground of the image, its body low to the ground as it **dashes towards the right side** of the frame. The antelopes, with their long, **curved horns**, are in hot pursuit. They are spread out across the field, their bodies leaning forward in an attempt to escape from the cheetah. The savanna stretches out behind them, a sea of green grass. The scene is a snapshot of life in the wild, a testament to the beauty and power of nature.

Figure 10: VLSA provides more detailed, visually-grounded responses than LLaVA-Next.

A.15 Training Efficiency Analysis

To address concerns about training cost, we provide a detailed breakdown of wall-clock training time on 8×NVIDIA A100 80GB GPUs:

Table 41: Wall-clock training time comparison.

Method	Stage I	Stage II	Total
LLaVA-Next	∼5.3h	∼30.0h	**35.3h**
VLSA	∼4.8h	∼14.1h	**18.9h (-47%)**

Time reduction analysis:

- **Reduced LLM computation (primary factor):** SA-Perceiver compresses visual tokens from ∼2880 to 576 (∼1/5), substantially lowering LLM self-attention cost in Stage II. Since the LLM dominates training compute, fewer tokens yield major wall-clock savings.
- **SD3 overhead:** Single-step denoising per iteration; the denoising transformer (∼2B parameters) adds modest overhead relative to LLM savings.
- **VAE overhead:** The VAE encoder (∼85M parameters) is frozen throughout training. It can optionally be pre-encoded offline for zero training cost (our experiments use online encoding).
- **VQ-VAE overhead:** VQ-VAE is used only for offline annotation (pre-computing codebook indices as SSFT labels)—zero training overhead.

Inference overhead: During inference, SD3 VAE, denoising transformer, and VQ-VAE are completely removed. VLSA's inference architecture consists only of CLIP + SA-Perceiver + LLM, which is more lightweight than baseline due to fewer visual tokens.

A.16 Attention and Representation Diagnostics

A.16.1 Grad-CAM Attention Analysis

We use Grad-CAM (Selvaraju et al., 2017) to examine how Cognition Alignment changes the image regions that contribute to answer generation. We compare full VLSA with an ablation that retains the same Perception Alignment components and reconstructive objective but omits Cognition Alignment. Because the two variants are trained separately, their learned visual representations need not be identical. The Grad-CAM comparison therefore characterizes their resulting end-to-end visual attribution patterns rather than isolating LLM utilization under fixed visual inputs. (Appendix A.17 provides that complementary control by feeding the same visual tensors to both LLMs.)

Method: For each image-question pair, we append the ground-truth answer token to the input sequence, perform forward propagation, and compute the gradient of the answer token's logit with respect to visual tokens in the LLM's intermediate layers. The gradient magnitude serves as the importance weight for each visual token, which we map back to image space to generate Grad-CAM heatmaps.

Quantitative analysis: We compute the entropy of the normalized gradient distribution over 1,000 images from the validation set:

$$H = -\sum_i p_i \log p_i \tag{11}$$

where p_i is the normalized gradient magnitude for visual token i. Lower entropy indicates more focused attention.

Qualitative visualization: Figure 11 illustrates the more concentrated gradient maps of full VLSA in three examples, complementing the entropy measurements.

Table 42: LLM visual attention entropy analysis.

Model	CAM Entropy	Interpretation
VLSA w/o CA	6.206	Dispersed attention
VLSA Full	**4.483**	Focused attention (-27.8%)

A.16.2 Centered Kernel Alignment (CKA) Analysis

To quantify vision-language semantic alignment within the LLM, we use Centered Kernel Alignment (CKA) (Kornblith et al., 2019) to measure the similarity between visual token representations and language token representations across different LLM layers.

Method: We select 1,000 images with ground-truth captions from COCO-Cap (Chen et al., 2015). For each image, we feed [image + ground-truth caption] to the model and extract hidden states from LLM layers {4, 8, 12, 16, 24, 32}. After mean-pooling tokens per image per layer, we compute linear CKA between visual representation matrix $V \in \mathbb{R}^{N\times d}$ and language representation matrix $L \in \mathbb{R}^{N\times d}$:

$$\text{CKA}(V, L) = \frac{\|L^\top V\|_F^2}{\|V^\top V\|_F \cdot \|L^\top L\|_F} \tag{12}$$

A higher CKA means that the relationships among examples in the visual representation space more closely match those in the corresponding language representation space. We therefore use CKA to assess representation-level vision-language alignment. The correct-versus-shuffled visual-representation experiment in Appendix A.17 separately evaluates whether Cognition Alignment improves the LLM's use of image-specific information when both LLMs receive identical visual tensors.

Table 43: Vision-language semantic alignment across LLM layers (CKA).

LLM Layer	Baseline	VLSA w/o CA	VLSA Full	Improvement
Layer 4	0.29	0.34	0.30	+0.01
Layer 8	0.26	0.31	**0.42**	+0.16
Layer 12	0.34	0.45	**0.57**	+0.23
Layer 16	0.54	0.52	**0.55**	+0.01
Layer 24	0.82	0.89	**0.88**	+0.06
Layer 32	0.79	0.77	**0.85**	+0.06

Key findings:

- Full VLSA has higher CKA at Layers 8–12 than both comparison variants, indicating greater vision-language representation similarity at these depths.
- At Layer 12, CKA increases from 0.34 for the baseline to 0.45 after Perception Alignment, and then to 0.57 when Cognition Alignment is added. Thus, both alignment stages are associated with higher vision-language representation similarity at this layer, with Cognition Alignment providing a further increase of 0.12 over the w/o-CA variant.
- The improvement is not uniform across layers: full VLSA has slightly lower CKA than the w/o-CA variant at Layers 4 and 24. We therefore interpret the result as stronger alignment concentrated in intermediate layers rather than a consistent increase at every layer.

A.17 Controlled Recoverability and Utilization Protocols

To complement the reconstruction analyses in Appendix A.1.4 and the Grad-CAM and CKA diagnostics in Appendix A.16, we conduct two controlled experiments: Shared-decoder reconstruction on unseen images examines information recoverable from visual embeddings (E-IL), and Correct-versus-shuffled evaluation with fixed visual representations tests the LLM's use of image-specific information already present in those representations (D-IL).

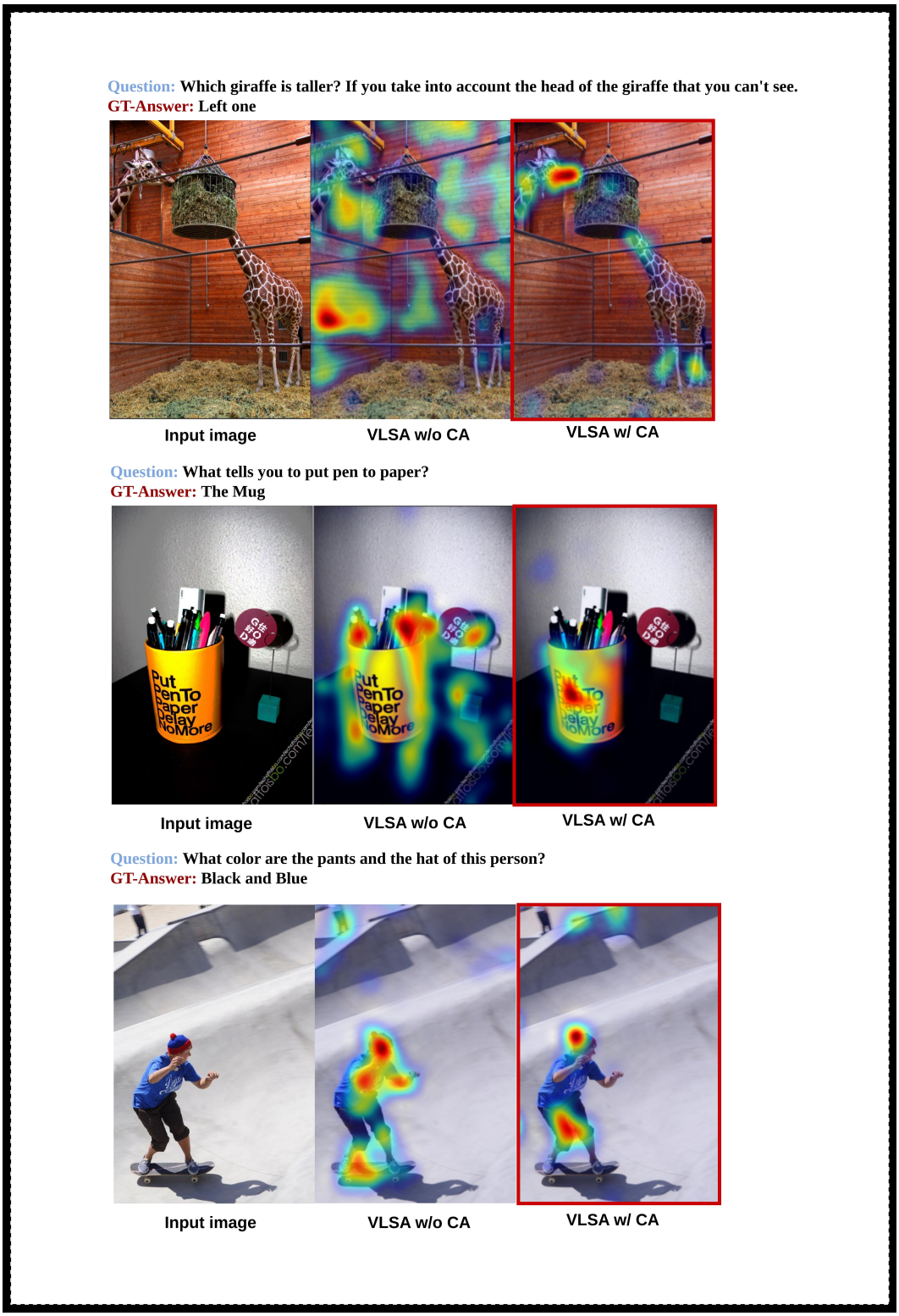

Figure 11: Grad-CAM visualization comparing VLSA w/o Cognition Alignment (dispersed attention) vs. VLSA Full (focused attention). Red regions indicate high attention weights. VLSA Full consistently attends to semantically relevant image regions, while VLSA w/o CA shows dispersed attention patterns.

Shared-decoder reconstruction on unseen images. To compare embedding recoverability under identical reconstruction conditions, we use a single decoder and a disjoint evaluation set. For both LLaVA-Next and VLSA, we freeze the vision encoder and alignment module and extract the resulting LLM-input visual embeddings for every image used to train the decoder. Each image is paired with one embedding from each model, and the two pairs are weighted equally during decoder training. The decoder is initialized from the official pretrained SD3 weights, and all reconstruction-side parameters, including the conditioning projections, are shared between the two embedding sources. At evaluation, the trained decoder is held fixed and applied without further adaptation to both embeddings of the same 1,000 Q-Bench (Wu et al., 2023) images, none of which appeared in VLSA or decoder training.

Table 8 reports a shared-decoder PSNR of 12.2 dB for LLaVA-Next and 21.8 dB for VLSA. Because both conditions use the same held-out images and fixed decoder, this gap cannot be attributed to separately optimized decoders or evaluation on training images. Instead, it shows that substantially more input-specific information is recoverable from VLSA's visual embeddings, providing direct operational evidence of lower E-IL under this controlled setting.

Correct-versus-shuffled V utilization gap. To test decoding-stage utilization directly, we randomly sample 500 Q-Bench questions. For each question q_i, we use the checkpoint trained without Cognition Alignment (w/o CA) to extract the representation V_i of its corresponding image. We feed that exact tensor to two LLMs: one from the checkpoint trained with Cognition Alignment (w/ CA) and the other from the w/o CA checkpoint. Both LLMs therefore receive identical visual information. We then generate three independent permutations π_k of the image assignments and use the same three permutations for both LLMs. Under permutation k, q_i is paired with $V_{\pi_k(i)}$ from a different image ($\pi_k(i) \neq i$), while its question, answer options, and prompt remain unchanged. Each LLM is thus evaluated once with V_i and three times with $V_{\pi_k(i)}$.

For LLM j, we define the correct-option log-odds S_j and per-example utilization gap $U_{i,j}$. We then compute ΔU by averaging the paired difference in U between w/ CA and w/o CA over all 500 questions:

$$S_j(q_i, V; c_i) = \log p_j(c_i \mid q_i, V) - \log \sum_{r \neq c_i} p_j(r \mid q_i, V), \tag{13}$$

$$U_{i,j} = S_j(q_i, V_i; c_i) - \frac{1}{3}\sum_{k=1}^{3} S_j(q_i, V_{\pi_k(i)}; c_i), \tag{14}$$

$$\Delta U = \frac{1}{500}\sum_{i=1}^{500} \left(U_{i,\mathrm{CA}} - U_{i,\mathrm{noCA}}\right). \tag{15}$$

Here, c_i is the correct option and r ranges over the incorrect options. We refer to $S_j(q_i, V_i; c_i)$ as the matched-image score. For permutation k, $S_j(q_i, V_{\pi_k(i)}; c_i)$ is the shuffled-image score, whose average over the three shared permutations gives the mean shuffled-image score. Their difference $U_{i,j}$ measures how much LLM j benefits from the correctly matched visual information for question i, and ΔU compares this benefit between the two LLMs under identical visual representations.

Table 8 reports means over the same 500 questions. The w/o CA and w/ CA LLMs obtain matched-image scores of -0.22 and 0.08, mean shuffled-image scores of -0.80 and -0.79, and utilization gaps of 0.58 and 0.87, respectively. The corresponding paired differences and question-level paired-bootstrap 95% confidence intervals are $+0.30$ $[0.24, 0.36]$, $+0.01$ $[-0.02, 0.04]$, and $\Delta U = +0.29$ $[0.23, 0.35]$. On the same questions with correctly matched images, accuracy increases from 58.4% (292/500) to 65.2% (326/500).

Because both LLMs receive identical V and shuffles, ΔU reflects their difference in decoding-stage use of task-relevant information already present in V, rather than encoding quality or an overall score advantage that persists after shuffling. Its 95% confidence interval excludes zero, providing direct operational evidence of lower D-IL under fixed visual representations.

Together, these controls provide direct operational evidence of lower E-IL and D-IL under the tested settings. Shared-decoder reconstruction holds the decoder and evaluation images fixed, while the fixed-V comparison removes encoding quality as an explanation for the utilization gap. These tests assess recoverability and utilization under the specified controls, without establishing a universal causal decomposition of information loss.

A.18 Matched Baselines and Original Model Context

Table 44 reports the original published results alongside our matched baselines and their VLSA variants. For Qwen2.5-VL and InternVL-2.5, we initialize each baseline and its VLSA variant from the same officially released checkpoint and further fine-tune both on LLaVA-Next's 790K instruction-tuning set. For SliME, whose original in-house training data are unavailable, we train both variants from the same initialization using LLaVA-Next's training data. Within each architecture, both variants follow the same training and evaluation protocols. We therefore assess VLSA's contribution by comparing the "Matched baseline" and "+VLSA" rows. The "official" rows report published performance under the original models' training and evaluation settings and serve as contextual references.

Table 44: Published model results and matched baseline/+VLSA comparisons. Dashes indicate unavailable published scores.

Backbone/setting	GQA	TextVQA	MMB	MME	MM-Vet	LLaVA-W	MMMU
Qwen2.5-VL official	–	84.9	83.5	2347	67.1	–	58.6
Matched baseline	68.7	84.1	82.8	2303	62.5	74.7	58.6
+VLSA	69.0	86.2	85.8	2354	66.3	74.8	61.4
InternVL-2.5 official	–	79.1	84.6	2344.1	62.8	–	56.0
Matched baseline	78.2	77.4	84.3	2322	64.9	78.5	53.8
+VLSA	78.2	79.6	84.8	2358	66.1	79.4	56.1
SliME official	63.94	64.76	75.00	1915	37.40	73.90	41.20
Matched baseline	64.6	64.1	70.9	1861	35.4	70.2	42.6
+VLSA	65.3	64.9	74.6	1887	38.0	70.8	46.0

A.19 Effect of Reconstruction Initialization

To assess the benefit of SD3 pretraining, we compare VLSA variants that use pretrained or randomly initialized SD3 for reconstructive training (Table 45). Pretrained initialization improves GQA, SQA-I, and DocVQA at both 336^* and 672×672 snapshot resolutions. At 672×672, the gains reach 3.5, 3.8, and 2.0 points, respectively, exceeding those at 336^*. Since both variants receive reconstruction supervision, this comparison evaluates the overall benefit of pretraining, including SD3's learned text-to-image conditioning.

To examine whether these gains are accompanied by stronger vision-language alignment, we compute centered linear CKA between visual and language representations within the LLM (Eq. 12). We use the same 1,000 COCO-Cap image–caption pairs and six LLM layers as in the LDM–MAE comparison (Table 29). The VLSA variant trained with pretrained SD3 has higher CKA at five of the six layers, with the largest increase at Layer 12 (Table 46). The smaller CKA differences at shallow layers may arise because visual and textual features in both variants still retain largely modality-specific semantics, resulting in similarly low CKA values. At deep layers, both variants already produce highly similar representations for images and their paired captions, so their CKA values are similarly high and differ little. The effect of pretrained SD3 may therefore be most visible at intermediate layers, where cross-modal alignment is still developing.

Table 45: SD3 initialization at two snapshot resolutions.

Res.	Init.	GQA	SQA-I	DocVQA
336^*	Scratch	65.0	76.3	74.7
336^*	Pretr.	65.3	77.5	75.2
672	Scratch	65.7	77.0	78.1
672	Pretr.	69.2	80.8	80.1

Table 46: CKA: pretrained versus scratch SD3.

Layer	Scratch	Pretr.	Δ
4	0.32	0.30	-0.02
8	0.37	0.42	$+0.05$
12	0.36	0.57	$+0.21$
16	0.49	0.55	$+0.06$
24	0.85	0.88	$+0.03$
32	0.82	0.85	$+0.03$

A.20 VISION-CENTRIC EVALUATION

We evaluate V*Bench (Wu & Xie, 2024) and MMVP (Tong et al., 2024b) to further assess performance on tasks that depend on visual detail. Table 47 uses a matched LLaVA-1.5 baseline, complementing the LLaVA-1.5-HD and LLaVA-Next comparisons in Table 9. Each result is interpreted relative to its own backbone and evaluation setting.

A.21 FIDELITY OF AUXILIARY PREDICTION TARGETS

We assess whether each auxiliary objective improves the fidelity of its predictions. For codebook indices, two variants differing in codebook-index supervision are evaluated on the same 1,000 image regions, with both sets of predictions decoded by the same frozen VQ-VAE codebook and decoder. For pixel prediction, we compare predicted RGB values with the original regions over 1,000 pairs, using variants trained with and without that objective. The PSNR gains in Table 48 show more faithful intermediate predictions in both comparisons. Together with the downstream ablations, they support the usefulness of these targets.

Table 47: Vision-centric results with LLaVA-1.5 and +VLSA.

Method	V*Bench	MMVP
LLaVA-1.5	48.68	24.70
+VLSA	52.77	29.60
Gain	+4.09	+4.90

Table 48: Auxiliary prediction PSNR (dB), evaluated on 1,000 region pairs.

Objective	Without	With	Gain
Codebook	8.9	14.3	+5.4
Pixel	10.2	13.4	+3.2