BRIDGING COMPRESSED IMAGE LATENTS AND MULTIMODAL LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents the first-ever study of adapting compressed image latents to suit the needs of downstream vision tasks that adopt Multimodal Large Language Models (MLLMs). MLLMs have extended the success of large language models to modalities (e.g. images) beyond text, but their billion scale hinders deployment on resource-constrained end devices. While cloud-hosted MLLMs could be available, transmitting raw, uncompressed images captured by end devices to the cloud requires an efficient image compression system. To address this, we focus on emerging neural image compression and propose a novel framework with a lightweight transform-neck and a surrogate loss to adapt compressed image latents for MLLM-based vision tasks. Given the huge scale of MLLMs, our framework has the striking feature of excluding the entire downstream MLLMs from training our system. This stands out from most existing coding for machine approaches that involve downstream networks in training and thus could be impractical when the networks are MLLMs. The proposed framework is general and applicable to various MLLMs and multiple application scenarios, where the neural image codec can be (1) pre-trained for human perception without updating, (2) fully updated for joint human and machine perception, or (3) fully updated for only machine perception. Extensive experiments on different neural image codecs and various MLLMs show that our method achieves great rate-accuracy performance with much less complexity, demonstrating its effectiveness.

1 INTRODUCTION

Large Language Models (LLMs) (Touvron et al., 2023b;a) have demonstrated impressive abilities in various Natural Language Processing (NLP) tasks. Building upon their success, the recent surge of research on Multimodal Large Language Models (MLLMs) extends LLM's abilities to modalities beyond languages, particularly images, opening up promising opportunities in various applications (Achiam et al., 2023; Cha et al., 2024; Lin et al., 2024; Zhang et al., 2024a). MLLMs have shown surprising capability for many vision tasks such as classification (Zhu et al., 2024b), image captioning (Zhang et al., 2024a), Visual Question Answering (VQA) (Cha et al., 2024; Lin et al., 2024), and meme interpretation (Achiam et al., 2023). These models excel in unseen tasks through instruction following or in-context learning, which is impossible for traditional vision networks.

However, the massive scale of MLLMs, often comprising billions of parameters, poses significant challenges for deployment on resource-constrained end devices. While computation can be offloaded to the cloud, transmitting images to cloud-hosted MLLMs becomes necessary. In this case, efficient compression techniques are crucial to reducing the bit-rate required for transmission. Without compression, transmitting raw images incurs significant costs, particularly at scale with numerous users. Our study shows that simply feeding the decoded image, generated by a fixed image codec trained for human perception, into an MLLM (Figure 1 (a)) substantially degrades task performance, particularly when the image is coded at low rates. This highlights the critical need for efficient image compression that considers the requirements of downstream MLLM-based vision tasks.

To the best of our knowledge, there have been no attempts to tackle image compression specifically for MLLMs. While many prior works address image compression for machine vision, commonly referred to as coding for machines (Le et al., 2021b; Chamain et al., 2021; Matsubara et al., 2022; Liu et al., 2022b; Le et al., 2021a), these approaches cannot be directly applied to MLLMs. Two

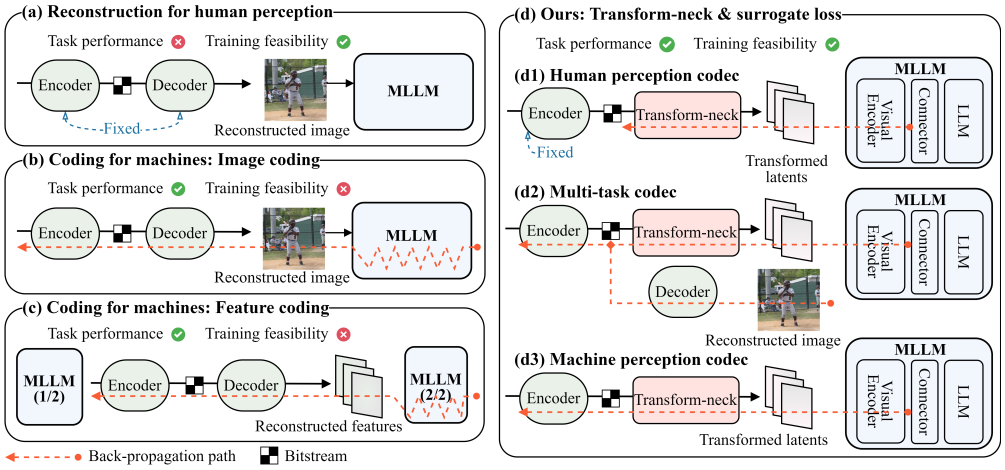


Figure 1: On the left is inadequate frameworks for image compression for MLLMs, where the image codec is trained for (a) human perception, (b) the downstream task network, or (c) compressing the intermediate features of the task network. On the right is the proposed transform-neck and surrogate loss under three distinct scenarios, with the image codec (d1) pre-trained for human perception, (d2) updated for joint human and machine perception, or (d3) updated for machine perception.

common approaches to coding for machines are image coding and feature coding. The image coding approaches (Le et al., 2021b;a) optimize the image codec for specific downstream tasks and/or networks (Figure 1 (b)), while the feature coding approaches (Ding et al., 2024) divide the task network into two parts and focus on compressing the intermediate features (Figure 1 (c)). However, both approaches face the same issue: the training process becomes challenging when one needs to back-propagate a training objective through a massive MLLM to train the neural image codec. In practice, the billion-scale parameters of MLLMs make the existing coding for machine methods inapplicable. Moreover, most existing methods limit the trained system to be only suitable for one specific model or task, thus requiring separate parameters or models for different tasks.

In this paper, we propose the first neural image compression system for MLLM-based vision tasks that enables compressed image latents to suit the needs of downstream MLLMs. The proposed method involves a lightweight transform-neck and a novel surrogate loss. The transform-neck adapts compressed image latents to match the intermediate features of the visual encoder, one of the components of MLLMs, bypassing the needs for full image decoding and reducing computational complexity. Our proposed surrogate loss function, which combines cross-entropy and distillation terms, enables our system to be trained by back-propagating solely through the visual encoder, thus eliminating the necessity of back-propagation through the entire MLLM.

The proposed method is general in that it is applicable to different neural image codecs under various application scenarios. First, if the downstream applications prioritize image reconstruction quality for human interaction, our method can work with an off-the-shelf image codec trained for human perception (Figure 1 (d1)). Without any modification or re-training of the codec, our method adapts the compressed image latents while maintaining the same image reconstruction quality. Second, when allowing the image codec to be updated, we propose a multi-task training strategy that optimizes the codec for both human and machine perception (Figure 1 (d2)). This significantly improves MLLM performance at the cost of a marginal drop in reconstruction quality. Finally, we consider an extreme setting in which the applications prioritize machine perception over image reconstruction. In this case, the encoder and the transform-neck are jointly optimized for the MLLM systems exclusively (Figure 1 (d3)). Our proposed framework is broadly applicable to a wide range of neural image codecs and MLLMs, regardless of their architectures. Additionally, the transform-neck trained with our surrogate loss exhibit a degree of universality, since it is readily applicable to multiple MLLMs that share the same visual encoder, without the need for retraining.

The contributions of this work are summarized as follows:

- It marks the first exploration into the field of neural image coding for MLLMs.
- The proposed transform-neck adapts the compressed image latents to downstream MLLMs, avoiding the need for image decoding and saving computational complexity.

- The proposed surrogate loss leverages the visual encoder to update the system, avoiding back-propagating the task loss through the heavy MLLM.
- Our method is applicable to different downstream MLLMs and neural image codecs, demonstrating its versatility.
- Our framework is able to accommodate various application scenarios that involve human perception, machine perception, or both.

2 Related Works

2.1 Multimodal Large Language Models

In recent years, there has been a surge of interest in MLLMs following the impressive demonstration of LLM's ability in the NLP field (Touvron et al., 2023b;a; Jiang et al., 2023). Many have sought to extend the success of these models from text to other modalities, particularly images, and several works have shown their effectiveness on various tasks, such as image captioning (Li et al., 2023b; Lin et al., 2024; Liu et al., 2023a; Chen et al., 2023a), VQA (Cha et al., 2024; Zhang et al., 2024a; Chen et al., 2023a), Referring Expression Comprehension (REC) (Chen et al., 2023b; Peng et al., 2024; Zhang et al., 2024b; You et al., 2024), few-shot classification (Yu et al., 2024; Zhu et al., 2024b), action anticipation (Mittal et al., 2024), meme interpretation (Achiam et al., 2023; Yang et al., 2023), OCR-free math reasoning (Driess et al., 2023).

Most existing MLLM approaches use a visual encoder to process the input image data, and then introduce a connector to bridge the image features to the tokens understandable by the LLM. Earlier works adopt simpler connector designs, such as linear projectors (Chen et al., 2023b; Liu et al., 2023a), while subsequent works (Li et al., 2023b; Cha et al., 2024; Zhang et al., 2024a) have refined upon the design for both performance and complexity. Furthermore, the entire MLLM can be further fine-tuned to enhance its capabilities through instruction tuning (Liu et al., 2023a; Zhu et al., 2024a).

A notable aspect of the MLLMs is their reliance on existing pre-trained visual encoders in their systems, with CLIP (Radford et al., 2021) visual encoder being a very common choice for a large number of methods (Cha et al., 2024; Chen et al., 2023b; Lin et al., 2024; Zhang et al., 2024a; Zhu et al., 2024b; Li et al., 2023b). Trained on large image-text pair data, the CLIP visual encoder offers the feature space that combines language and image modalities in a sense, making it a desirable feature for MLLMs. Notably, all the existing works on MLLMs do not consider the scenarios where image compression is present, which is a significant departure from our work. We note that some other approaches (Shi et al., 2024; Li et al., 2024) reduce the number of tokens for MLLM efficiency, which is orthogonal to our method. A detailed discussion is provided in Section A.6 of the supplementary.

2.2 Image Coding for Machines

Neural image compression systems have made significant progress in the past few years. As a matter of fact, several works (He et al., 2022; Lu et al., 2022a; Liu et al., 2023b; Xie et al., 2021) have even outperformed the traditional codecs such as intra coding in VVC (Bross et al., 2021). However, these methods primarily focus on the quality of reconstructed images for human perception. Coding for machines, in contrast, targets downstream machine vision instead of human perception, and it has attracted increasing attention recently.

A common approach simply involves training the compression system for a predefined target downstream computer vision task (Le et al., 2021b;a; Wang et al., 2022), enabling the reconstructed image to be suitable for machine vision, albeit potentially sacrificing perceptual quality. Conversely, Chamain et al. (2021) tune the task network to better process the compressed images, while Chen et al. (2023c) leverage prompt-tuning method on Transformer-based codecs to boost performance on multiple tasks. Also, with the trend of the new JPEG AI learning-based image coding standard (Ascenso et al., 2023), some methods (Liu et al., 2022a; 2021; Mei et al., 2021; Singh et al., 2020) utilize the compressed image latents instead of the reconstructed image for recognition through bridging the latents to task network. On the other hand, (Ding et al., 2024) directly compress the intermediate features of recognition networks, while (Feng et al., 2022) learn the omnipotent features suitable for various tasks in a self-supervised manner and fine-tune each task network tail on such features.

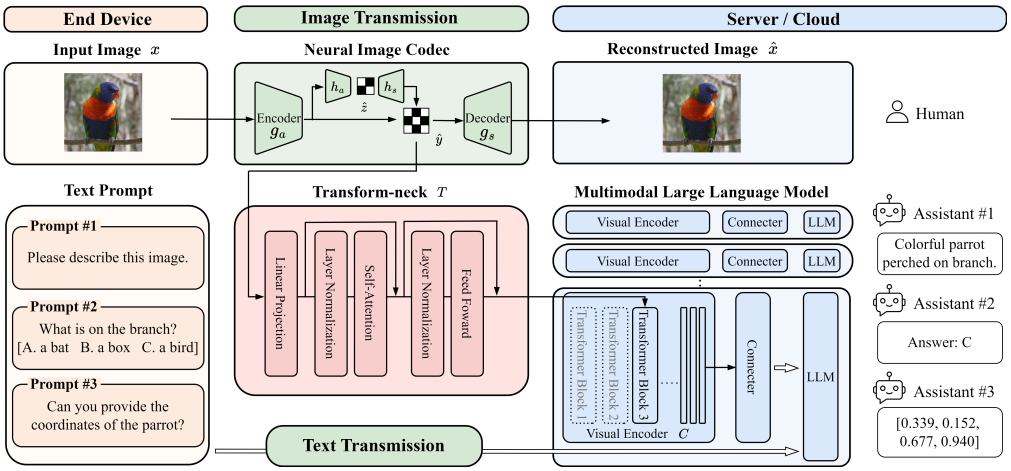


Figure 2: Overall architecture of the proposed method.

It is crucial to note that none of the coding for machine methods considers MLLMs at the receiver side. All the above-mentioned methods leverage back-propagation through recognition models to update the system or even re-train the recognition network itself, both of which are prohibitively expensive for MLLMs due to their huge scale. Therefore, the direct application of the same methods on MLLMs is almost infeasible. In addition, mainstream image coding for machines methods, including JPEG AI (Ascenso et al., 2023), remain mostly task-specific. They typically adopt a task-based loss, which restricts the resulting models to be optimized for a single task and recognition model, thus requiring re-training for each new task and incurring additional costs. We aim to be the first to propose a neural image compression system designed for MLLMs, achieved through a lightweight transform-neck and the adoption of a surrogate loss, which allows to bypass the necessity of involving the entire billion-scale MLLM in the training process. Note that our surrogate loss incorporates a cross-entropy loss to bridge visual features with the text domain for MLLMs, complementing the feature-constraining distillation loss. This combination further differentiates our approach from existing methods.

3 Proposed Method

3.1 Preliminaries: Neural Image Codecs

The high-level architecture of a neural image codec is depicted in the top central green box in Figure 2. In a typical hyperprior-based neural image compression system (Ballé et al., 2018), the key components include the main encoder g_a, the main decoder g_s, as well as the hyperprior encoder h_a and decoder h_s. Given an RGB image $x \in \mathbb{R}^{3\times H\times W}$, where H and W represent the height and width of the image, respectively, g_a performs the analysis transform of x and generates the image latent representation $y \in \mathbb{R}^{N\times \frac{H}{16}\times \frac{W}{16}}$, with N indicating the channel size. To transmit y more efficiently, it is first uniformly quantized into $\hat{y}$ and then entropy coded considering a learned prior distribution $p(\hat{y})$. This learned distribution is content dependent, thanks to the hyperprior encoder h_a and decoder h_s. In particular, h_a takes y as input and produces the side information $z \in \mathbb{R}^{N_h\times \frac{H}{64}\times \frac{W}{64}}$, that is used to generate the learned distribution for entropy coding, where N_h is the channel size of the side information. The quantized version of z, denoted as $\hat{z}$, is transmitted into the bitstream, in order to recover $\hat{y}$. Lastly, $\hat{y}$ undergoes the synthesis transform with g_s, which reconstructs the image $\hat{x} \in \mathbb{R}^{3\times H\times W}$.

3.2 Overall Framework

In this work, we focus on the scenario where MLLMs are hosted on the server side, while users on end devices need to perform inference on the remote model using both text and images as inputs. Given the necessity of incorporating image compression to ensure efficient transmission, we propose the first compression framework with the consideration of MLLMs as downstream application networks, aiming to mitigate the potential task performance drop caused by image compression. Figure 2 illustrates our overall framework, which includes three major components: the neural image

codec, our proposed transform-neck, and the MLLM. The depicted MLLM system adheres to a typical structure, consisting of a visual encoder, an LLM, and a connector component facilitating the transformation of features from the visual encoder to the LLM. Note that all the MLLMs are adopted off-the-shelf and without any update.

During inference, the input image at the end device is passed through an encoder g_a to generate the quantized latents $\hat{y}$ for transmission. Next, $\hat{y}$ is directly passed through a lightweight transform-neck T for transformation into a middle layer of the visual encoder of an MLLM. We opt to adapt the image latents rather than the reconstructed images because the image latents inherently contain the information needed for reconstructing the image, and potentially the semantic information for the downstream tasks (when the image encoder is guided properly). By skipping the image decoding process, our method offers reduced computational complexity while maintaining the task performance. The rest of the MLLM system operates without any changes to generate the desired output response.

In training, to address the challenge of back-propagating the task loss through the entire MLLM, we propose a novel surrogate loss that updates the system by back-propagating solely from the visual encoder (which is not re-trained), bypassing the billion-parameter LLM. In our work, we examine three distinct settings denoted as (d1), (d2) and (d3), as illustrated in Figure 1. Firstly, in (d1), we consider the practical scenario where a fixed off-the-shelf image codec pre-trained for human perception is directly used alongside our transform-neck. In this setting, our framework offers the option for users to decode the image latents $\hat{y}$ for reconstruction by using the decoder g_s instead of the transform-neck. In this way, the quality of the decoded image is not affected, as the image codec is not updated in the present case. Then, we extend the analysis to scenarios (d2) and (d3) to examine the impact of jointly training the image codec and transform-neck. In (d2), the entire image codec undergoes re-training to accommodate both human and machine perception, while in (d3), the encoder is re-trained specifically for machine perception.

3.3 TRANSFORM-NECK

Our transform-neck is designed to be a lightweight module, consisting only of a linear projection, a self-attention mechanism, a feed-forward layer, and two layer norms, as shown in the central red box in Figure 2. Its purpose is to adapt the compressed image latents $\hat{y}$ into an efficient representation for consumption by the downstream MLLMs. In fact, rather than reconstructing the image and using it as input to the MLLM, we propose leveraging the latent representation directly.

Since the image encoder g_a already functions as a feature extractor, similar to the early layers of the visual encoder C, we bridge the output of our transform-neck directly into the intermediate features of C, effectively integrating the image codec with the MLLM system. The decision on which initial layers to bypass depends on the specific type of visual encoder used in the MLLM. For instance, when using the CLIP visual encoder as C, we found that connecting the transform-neck to the third Transformer layer, bypassing the first two, yields optimal results (see Section 4.5 for ablation experiments justifying this design). Note that skipping the initial layers of C further reduces computational complexity of our framework. We denote the partial visual encoder as C'.

3.4 SURROGATE LOSS

To avoid involving huge MLLMs in the training process, and thus bypassing back-propagation through their entire structure, we propose a surrogate loss $\mathcal{L}_S$ which is back-propagated through only the partial visual encoder C'. Specifically, we design our surrogate loss to consist of two terms: distillation loss $\mathcal{L}_{dist}$ and cross-entropy loss $\mathcal{L}_{CE}$.

To retain the downstream MLLM performance, the resulting features $C'(T(\hat{y}))$ when using our transformed latents should resemble closely those obtained when inputting the uncompressed image into C, that is $C(x)$. To this end, we introduce the following distillation loss $\mathcal{L}_{dist}$, aiming to minimize the Mean Squared Error (MSE) between the two output features:

$$\mathcal{L}_{dist} = \text{MSE}(C'(T(\hat{y})), C(x)). \tag{1}$$

In addition to the distillation loss, we propose incorporating a second term, which is the cross-entropy loss $\mathcal{L}_{CE}$ calculated using the following procedure. We take a classification dataset consisting of n images and m text labels $\{l_1, l_2, \ldots, l_m\}$, which correspond to m distinct classes. Each image

Table 1: Application scenarios for our method with corresponding training objective.

Application Scenario	Update Image Codec	Human Viewing	Phase 1 Loss Function	Phase 2 Loss Function
(d1) Human perception	✗	✓	$\mathcal{L}_S$	—
(d2) Multi-task	✓	✓	$\mathcal{L}_S$	$R + \lambda(\gamma d_{recon}(x, \hat{x}) + \delta \mathcal{L}_{dist})$
(d3) Machine perception	✓	✗	$\mathcal{L}_S$	$R + \lambda \mathcal{L}_{dist}$

belongs to one of these m classes. Next, we compute: (1) the n image embedding $C'(T(\hat{y}))$ and (2) the m text embeddings $C_t(l_j)$, where C_t is a text encoder applied to each text label l_j. In particular, we use the CLIP text encoder, independently of the visual encoder integrated into the MLLM under consideration. For each image, we then calculate the cosine similarity $\text{CS}(\cdot, \cdot)$ between its image embedding $C'(T(\hat{y}))$ and each of the m text embeddings $C_t(l_j)$ for $j = 1, 2, \ldots, m$. This produces a vector v of similarity scores between the given image and all the m classes, that is:

$$v = [\text{CS}(C'(T(\hat{y})), C_t(l_1)), \ldots, \text{CS}(C'(T(\hat{y})), C_t(l_m))]. \tag{2}$$

The resulting similarity vector is transformed into a probability distribution over the m classes using a softmax function. Finally, the cross-entropy loss is computed with respect to the corresponding one-hot encoded ground truth label vector $\mathbf{t}$, thus:

$$\mathcal{L}_{CE} = \text{CE}(\text{Softmax}(v), \mathbf{t}). \tag{3}$$

This approach aims to bridge the visual features to the text domain specifically for MLLM-based vision tasks, distinguishing our method from existing works in the field of coding for machines. We validate the effectiveness of our loss function design in the ablation study (Section 4.5).

3.5 Training Procedure

To explore the capabilities of our method under the application scenarios introduced in Section 3.2, we employ a two-phase training procedure that adapts to different scenarios. The first phase is shared for all scenarios and focuses on training the transform-neck exclusively. The second phase, applicable only to scenarios where the codec can be updated, involves joint optimization of both the transform-neck and the image codec. Table 1 provides a summary of the training procedure.

Phase 1: Transform-neck Training In the first phase, we train only the transform-neck using a progressive training strategy with the surrogate loss $\mathcal{L}_S$ and an off-the-shelf image codec. This phase is divided into three stages:

$$\mathcal{L}_S = \begin{cases} \mathcal{L}_{CE}, & \text{epoch} < E_1, \\ \alpha \mathcal{L}_{CE} + \beta \mathcal{L}_{dist}, & E_1 \leq \text{epoch} < E_2, \\ \mathcal{L}_{dist}, & \text{epoch} \geq E_2, \end{cases} \tag{4}$$

where the weighting factors α and β are set with a ratio of 1:100 for the two loss terms, and E_1, E_2 are empirically set to 20 and 40, respectively, in our experiments. This initial phase ensures that the transform-neck learns the transformation to align with the target latent space.

Phase 2: Joint Optimization For (d2) and (d3), where the image codec is allowed to be re-trained to produce latent representation more suitable for machine perception, the second phase is introduced with transform-neck and image codec jointly updated after phase 1 converges.

The scenario (d2), referred to as multi-task, aims to accommodate both human and machine perception. As a result, it is trained jointly with the transform-neck on both the distillation loss and traditional rate-distortion loss, i.e. $\mathcal{L}_{d2} = R + \lambda(\gamma d_{recon}(x, \hat{x}) + \delta \mathcal{L}_{dist})$, where $R = -\log p(\hat{z}) - \log p(\hat{y}|\hat{z})$ is the estimated rate of $\hat{y}$ and $\hat{z}$, and d_{recon} is the reconstruction loss calculated as the MSE between the uncompressed image x and the reconstructed image $\hat{x}$. The hyper-parameter λ controls the rate-distortion trade-off, while γ and δ weight the two losses.

In (d3), where the downstream applications do not require image reconstruction, the encoder and transform-neck are jointly optimized to minimize the trade-off cost between the rate R and the distillation loss $\mathcal{L}_{dist}$, thus disregarding the reconstruction quality. The training objective is thus $\mathcal{L}_{d3} = R + \lambda \mathcal{L}_{dist}$.

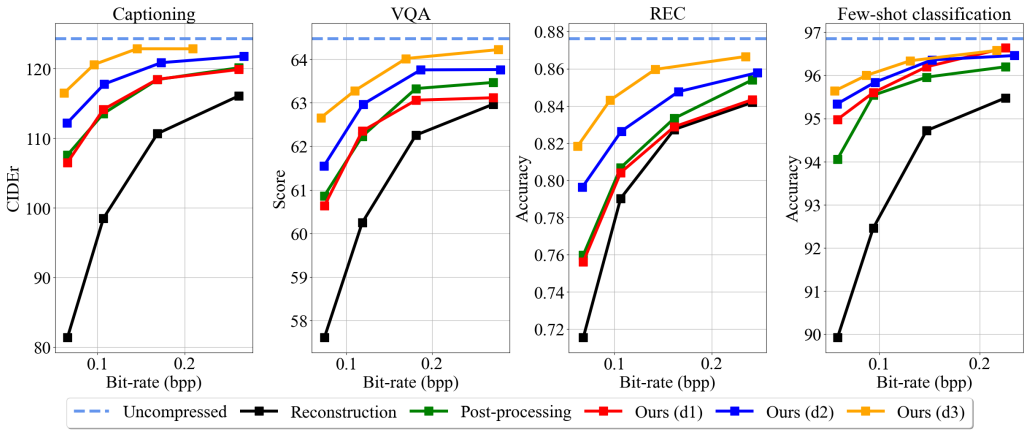


Figure 3: Rate-accuracy comparison using various MLLMs on several tasks.

Table 2: Evaluated tasks with corresponding dataset and MLLM.

Task	Dataset	MLLM
Captioning	COCO Karpathy Test (Karpathy & Fei-Fei, 2015)	LLaMA-Adapter (Zhang et al., 2024a)
VQA	SEED-Bench (Li et al., 2023a)	Honeybee (Cha et al., 2024)
REC	RefCOCO-val (Kazemzadeh et al., 2014)	Shikra (Chen et al., 2023b)
Few-shot classification	ImageNet (Deng et al., 2009)	V2L-Tokenizer (Zhu et al., 2024b)

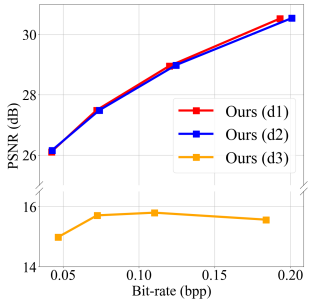


Figure 4: Reconstruction performance comparison on Kodak.

4 Experimental Results

4.1 Experimental Setting

Training Details and Datasets. We utilize ELIC (He et al., 2022) as our image codec, which outputs image and hyperprior latents with $N = 320$ and $N_h = 192$, respectively. ELIC is trained for human perception and adheres to the training strategy outlined in He et al. (2022), using 8,000 images of the highest spatial resolution selected from ImageNet dataset. Four models are trained for four different rate points, corresponding to $\lambda = [0.004, 0.008, 0.016, 0.032]$ in (He et al., 2022). For each of our scenarios (d1), (d2) and (d3), separate transform-necks are trained on ImageNet dataset (Deng et al., 2009) for individual λ values. For the scenario (d2) specifically, we find empirically that fixing the ratio $\gamma : \delta = 60 : 1$ leads to a good trade-off between human and machine perception. Given that most MLLMs adopt the pre-trained visual encoder of CLIP ViT-L/14 (Radford et al., 2021) for image modality, as discussed in Section 2.1, we use the CLIP visual encoder as C for training and conduct our primary experiments on MLLMs that incorporate it. It is worth noting that, since we consider MLLMs sharing the same visual encoder, we do not need to train separate systems for the different MLLMs or tasks. Furthermore, we provide additional experiments in Section 4.6 with MLLMs that use visual encoders other than CLIP ViT to demonstrate the generalizability of our approach.

Targeted MLLM-based Vision Tasks. To validate the generalization ability of our proposed method, we evaluate its performance on four different MLLM systems for four different tasks. The tasks, datasets, corresponding MLLMs, and metrics are listed in Table 2. These configurations follow the settings outlined in their original papers and the accompanying code, except for the few-shot classification task due to the inaccessibility of the code. We thus design a 5-way 1-shot classification scenario to evaluate the performance with in-context learning; the detailed setting is described in supplementary material. All MLLMs are used off-the-shelf without any fine-tuning.

Baselines. We introduce two baseline methods for comparison. The first one, denoted as *Reconstruction*, involves inputting the reconstructed image generated by ELIC to the MLLM system.

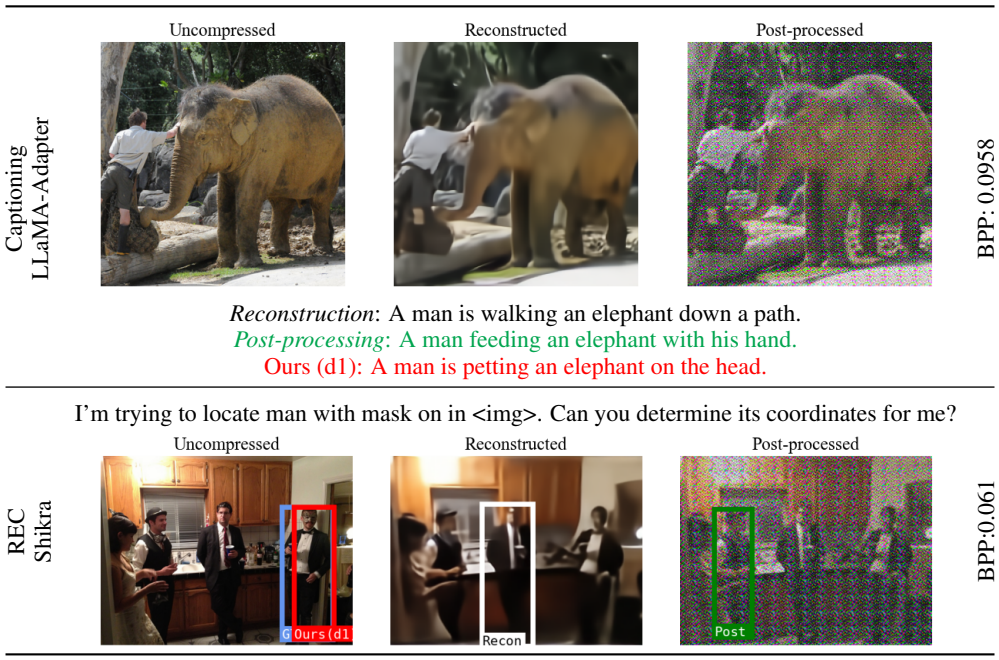


Figure 5: Visualization examples of our proposed method in (d1), *Reconstruction*, and *Post-processing* on image captioning with LLaMA-Adapter and REC with Shikra.

The second one, denoted as *Post-processing*, adapts the reconstructed image to MLLMs through a U-Net (Ronneberger et al., 2015) post-processing network, which is trained using the same surrogate loss as that adopted by our method. We remark that these image-domain baselines incur higher complexity than our lightweight transform-neck, as they involve decoding the image and potentially processing it further with the post-processing network.

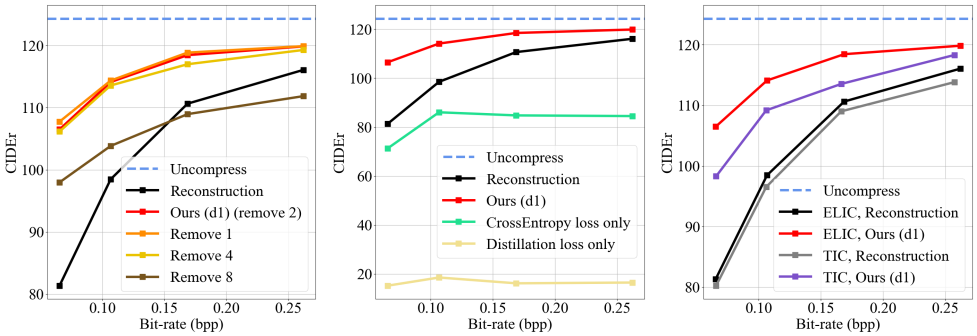
4.2 Performance Comparison

Figure 3 illustrates the performance of the baseline methods and our proposed scheme for the three examined scenarios with regards to two aspects: coding bit-rate, calculated as bits per pixel (bpp), and task performance. When comparing the baselines and our method in scenario (d1), where the original ELIC is trained solely for human perception, we make the following observations. (1) Straightforwardly using the reconstructed images generated by a codec trained for human perception leads to a significant performance drop across all the tasks (*Reconstruction*). Such performance decline is expected because the MLLMs are not trained with compressed images, thus hindering their recognition performance. This highlights the necessity of adapting image compression and/or image latents to MLLMs. (2) In contrast, our transform-neck method successfully boosts the performance using the same latent representations for reconstructing the image in *Reconstruction*, confirming the effectiveness of the proposed latent transformation without the decoding process. (3) *Post-processing* is able to reach comparable performance to our (d1), offering another viable solution to the problem. However, it is worth noting that *Post-processing* requires relatively higher computational complexity with respect to our transform-neck method, rendering our approach preferable (see Section 4.4).

Next, we evaluate the effects of allowing the image codec to be re-trained. First, we observe that (d2) outperforms both (d1) and *Post-processing*. This indicates that fine-tuning the encoder indeed results in a more suitable latent representation that can be better adapted to MLLMs. When examining the extreme setting (d3), we see significant further improvement in the task performance, approaching the performance upper bound with uncompressed images. This improvement comes at the cost of the image reconstruction quality, which, however, is not relevant in (d3). Figure 4 illustrates the rate-visual quality curves associated with the three scenarios. Interestingly, (d2) exhibits only a marginal PSNR drop compared to (d1), while (d3) significantly compromises the quality of the decoded image. We stress that our framework (i.e. the surrogate loss and transform-neck) is able to accommodate different application scenarios, allowing for a variable trade-off between the task performance and the image reconstruction quality.

Table 3: Comparison of the kMACs/pixel and model size. The table omits the shared components of the two methods, i.e. the image encoder, the partial CLIP visual encoder, the connector, and the LLM.

Method	Component	Params (M)		kMAC/pixel	
Ours (d1, d2, or d3)	Transform-neck	13.19		52.795	
Post-processing	Decoder Post-processing network First 2 layers of CLIP visual encoder	7.34 31.04 25.78	64.16 **(+386%)**	112.00 835.72 70.24	1017.96 **(+1828%)**



(a) Partial CLIP visual encoder (b) Training objectives (c) Different image codecs

Figure 6: Rate-accuracy comparison for three ablation studies evaluated with image captioning task.

4.3 Visualization of the Results

We present the visualization of outcomes with downstream MLLM-based vision tasks in Figure 5. Our method (d1) is compared with the two baseline methods, *Reconstruction* and *Post-processing*, with particular focus on how these models work at low bitrates to reflect a bandwidth-limited scenario. In the second and third columns, we visualize the reconstructed and post-processed images from the two baselines, respectively, which exhibit drastically different characteristics. The former (*Reconstruction*) produces blurry and smooth images, while the latter (*Post-processing*) introduces some artificial patterns into the post-processed images. Compared with the baselines, our method yields better MLLM results. More visualization are presented in Section A.3 of the supplementary material.

4.4 Complexity Analysis

Table 3 compares the computational complexity between our proposed method and *Post-processing* baseline in terms of model size and the kilo-multiply-accumulate-operations per pixel (kMACs/pixel). Note that our method in Table 3 refers to any of (d1), (d2), and (d3), since they share the same computational complexity characteristics at inference time. Our method offers a lightweight solution with only 13 million parameters, as opposed to 64 million parameters with the post-processing approach. Moreover, in terms of kMAC/pixel, the difference stands out even more, considering that the post-processing network operates at the full image resolution while our method operates in the latent domain, where the image latents have a much smaller spatial resolution.

4.5 Ablation Studies

The following ablation experiments are performed based on (d1) to justify our design choices.

Training Objective. Figure 6 (b) presents the performance of our method when trained exclusively with the cross-entropy loss or distillation loss. It is observed that training with only the cross-entropy loss results in a significant performance drop. Although providing a good initial update direction, this strategy is unable to learn an effective transformation for MLLMs. Instead, training solely with the distillation loss fails to update the transform-neck properly and leads to far inferior performance. This is potentially due to the stringent requirement of fitting the exact feature representations. Our proposed method, which is not a simple standard application of distillation, achieves highest performance.

To further support this finding, we present the following analysis in Figure 7: we first calculate the mean squared error (MSE) between CLIP visual encoder output tokens derived from uncompressed

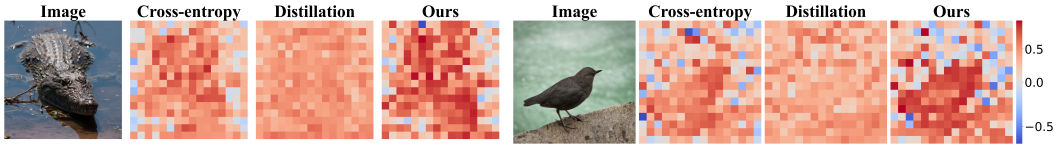


Figure 7: Visualization of MSE reduction on CLIP visual encoder output tokens before and after training using different loss functions. Darker red colors indicate greater MSE reduction.

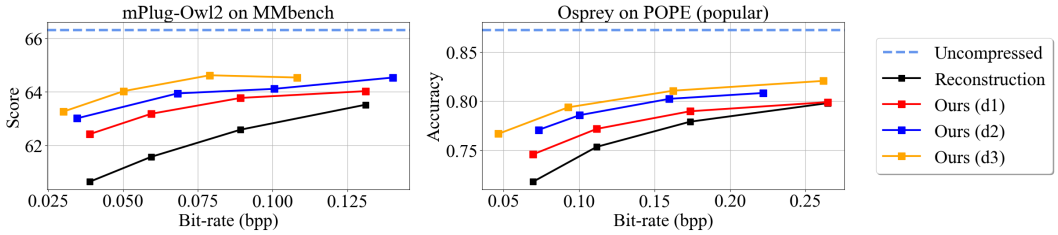


Figure 8: Rate-accuracy comparison on two MLLMs not utilizing CLIP ViT visual encoder: mPLUG-Owl2 (Ye et al., 2024) on MMBench and Osprey (Yuan et al., 2024) on POPE (popular setting).

images and from our transform-neck, both before and after the transform-neck has been trained. Then, we compute the difference between these MSEs to measure the improvement achieved by the specific training objectives all with equal training steps. Figure 7 shows that the cross-entropy loss reduces feature matching errors primarily in foreground object regions, while the distillation loss reduces global matching errors. Our proposed progressive training strategy integrates these two losses, leading to a greater reduction in MSE and thus improved rate-accuracy performance.

Partial CLIP Visual Encoder. This experiment investigates the proper number of Transformer layers to remove from the CLIP visual encoder in order to strike a good balance between complexity and performance. As shown in Figure 6 (a), removing the first one or two layers achieves similar performance, whereas removing four or eight layers results in a noticeable performance drop. We thus remove the first two layers.

Different Image Codecs. Figure 6 (c) presents the performance comparison between our method and *Reconstruction* when they are tested with ELIC and TIC (Lu et al., 2022a;b). TIC is a Transformer-based codec, whereas ELIC is a convolutional neural network-based codec. We see that our transform-neck still outperforms *Reconstruction* by a significant margin when the image codec is changed from ELIC to TIC. This indicates that our method is still effective on a different type of image codec.

4.6 Generalization

While we utilize the CLIP ViT visual encoder in our main experiments due to its wide popularity, our proposed method is applicable to various downstream MLLMs regardless of the visual encoder they adopt. As illustrative examples, Figure 8 presents the rate-accuracy performance of our re-trained scheme applied to two MLLMs that do not use the pre-trained CLIP ViT visual encoder: (1) mPlug-Owl2 (Ye et al., 2024) with a custom-trained ViT visual encoder, and (2) Osprey (Yuan et al., 2024) with a CLIP ConvNeXt-based visual encoder. Our method under all three settings clearly outperforms the *Reconstruction* baseline, confirming the generalizability of the proposed framework.

5 Conclusion

This paper proposes the first image compression system tailored to Multimodal Large Language Models (MLLMs). It introduces a transform-neck that bridges the compressed image latents and the intermediate layer of the visual encoder. By using our proposed surrogate loss, we avoid involving the entire MLLM in the training process and ensure downstream task performance. With lower computational complexity, our method has demonstrated effectiveness across a wide variety of tasks, MLLMs, and neural image codecs, outperforming other baselines in extensive experiments. One consideration is that this paper focuses solely on the image compression aspect, leaving the exploration of MLLM-based video or audio coding for future work.

REPRODUCIBILITY STATEMENT

We have included the information needed to reproduce the main experimental results in the paper. The architecture details are described in Section 3.3 and Figure 2, and the implementation details needed for reproduction is presented in Section 4.1 of the main paper and Section A.1 of the supplementary material. We also plan to release the code when the paper is made public for reproducibility.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Joao Ascenso, Elena Alshina, and Touradj Ebrahimi. The jpeg ai standard: Providing efficient human and machine visual data consumption. *Ieee Multimedia*, 30(1):100–111, 2023.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

Lahiru D Chamain, Fabien Racapé, Jean Bégaint, Akshay Pushparaja, and Simon Feltman. End-to-end optimized image compression for machines, a study. In *2021 Data Compression Conference (DCC)*, pp. 163–172. IEEE, 2021.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023a.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023b.

Yi-Hsin Chen, Ying-Chieh Weng, Chia-Hao Kao, Cheng Chien, Wei-Chen Chiu, and Wen-Hsiao Peng. Transtic: Transferring transformer-based image compression from human perception to machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23297–23307, 2023c.

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Ding Ding, Zhenzhong Chen, Zizheng Liu, Xiaozhong Xu, and Shan Liu. Hierarchical image feature compression for machines via feature sparsity learning. *IEEE Signal Processing Letters*, 2024.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023.

Ruoyu Feng, Xin Jin, Zongyu Guo, Runsen Feng, Yixin Gao, Tianyu He, Zhizheng Zhang, Simeng Sun, and Zhibo Chen. Image coding for machines with omnipotent feature learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*, pp. 510–528. Springer, 2022.

Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang. Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5718–5727, 2022.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

Eastman Kodak. Kodak lossless true color image suite (PhotoCD PCD0992). URL `http://r0k.us/graphics/kodak`.

Nam Le, Honglei Zhang, Francesco Cricri, Ramin Ghaznavi-Youvalari, and Esa Rahtu. Image coding for machines: an end-to-end learned approach. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1590–1594. IEEE, 2021a.

Nam Le, Honglei Zhang, Francesco Cricri, Ramin Ghaznavi-Youvalari, Hamed Rezazadegan Tavakoli, and Esa Rahtu. Learned image coding for machines: A content-adaptive approach. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2021b.

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*, 2023a.

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

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

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023c. URL `https://openreview.net/forum?id=xozJw0kZXF`.

Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023a.

Jinming Liu, Heming Sun, and Jiro Katto. Learning in compressed domain for faster machine vision tasks. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pp. 01–05, 2021. doi: 10.1109/VCIP53242.2021.9675369.

Jinming Liu, Heming Sun, and Jiro Katto. Improving multiple machine vision tasks in the compressed domain. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 331–337. IEEE, 2022a.

Jinming Liu, Heming Sun, and Jiro Katto. Improving multiple machine vision tasks in the compressed domain. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 331–337. IEEE, 2022b.

Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14388–14397, 2023b.

Ming Lu, Fangdong Chen, Shiliang Pu, and Zhan Ma. High-efficiency lossy image coding through adaptive neighborhood information aggregation. *arXiv preprint arXiv:2204.11448*, 2022a.

Ming Lu, Peiyao Guo, Huiqing Shi, Chuntong Cao, and Zhan Ma. Transformer-based image compression. In *Data Compression Conference*, 2022b.

Yoshitomo Matsubara, Ruihan Yang, Marco Levorato, and Stephan Mandt. Supervised compression for resource-constrained edge computing systems. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2685–2695, 2022.

Yixin Mei, Fan Li, Li Li, and Zhu Li. Learn a compression for objection detection - vae with a bridge. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pp. 1–5, 2021. doi: 10.1109/VCIP53242.2021.9675387.

Himangi Mittal, Nakul Agarwal, Shao-Yuan Lo, and Kwonjoon Lee. Can't make an omelette without breaking some eggs: Plausible action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. In *International Conference on Learning Representations (ICLR)*, 2024.

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.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Dachuan Shi, Chaofan Tao, Anyi Rao, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers. In *International Conference on Machine Learning (ICML)*, 2024.

Saurabh Singh, Sami Abu-El-Haija, Nick Johnston, Johannes Ballé, Abhinav Shrivastava, and George Toderici. End-to-end learning of compressible features. In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 3349–3353. IEEE, 2020.

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

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

Shurun Wang, Zhao Wang, Shiqi Wang, and Yan Ye. Deep image compression towards machine vision: A unified optimization framework. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

Yueqi Xie, Ka Leong Cheng, and Qifeng Chen. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM international conference on multimedia*, pp. 162–170, 2021.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13040–13051, 2024.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. In *International Conference on Learning Representations (ICLR)*, 2024.

Lijun Yu, Yong Cheng, Zhiruo Wang, Vivek Kumar, Wolfgang Macherey, Yanping Huang, David Ross, Irfan Essa, Yonatan Bisk, Ming-Hsuan Yang, et al. Spae: Semantic pyramid autoencoder for multimodal generation with frozen llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28202–28211, 2024.

Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. In *International Conference on Learning Representations (ICLR)*, 2024a.

Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. Groundhog: Grounding large language models to holistic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024b.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations (ICLR)*, 2024a.

Lei Zhu, Fangyun Wei, and Yanye Lu. Beyond text: Frozen large language models in visual signal comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024b.

A Supplementary Material

A.1 Implementation Details

Training. We use the Adam optimizer, configured with β_1 at 0.9, β_2 at 0.999, ϵ at 10^{-8}. Weight decay is disabled. The transform-neck for each rate point undergoes training on an RTX 4090 for approximately three days during the training stage.

Evaluation. For few-shot classification with V2L-Tokenizer (Zhu et al., 2024b), we design a 5-way 1-shot classification evaluation scenario. In particular, we generate 5000 groups of images from ImageNet dataset, where each group consists of five randomly sampled images from different classes, serving as the sample images, and one new image from one of the classes as the query image.

Different MLLM is utilized for the evaluation of our proposed method on each task. In Table 4, we provide the detailed specifications of the MLLM used in our evaluation.

Table 4: The specifications of the MLLM used in our tasks.

Task	Model	LLM
Captioning	LLaMA-Adapter v1 (Zhang et al., 2024a)	LLaMA-7B (Touvron et al., 2023a)
VQA	Honeybee-C-7B-M144 (Cha et al., 2024)	Vicuna-7B (Chiang et al., 2023)
REC	Shikra-7B (Chen et al., 2023b)	LLaMA-7B (Touvron et al., 2023a)
Few-shot classification	V2L-Tokenizer (Zhu et al., 2024b)	LLaMA2-7B (Touvron et al., 2023b)

A.2 Comparison with VVC

Figure 9 compares *Reconstruction* and our method in (d1) using ELIC, with the state-of-the-art traditional codec VVC (VTM 17.0 intra coding). We set the QPs to $[37, 40, 43, 46, 49]$ for VVC. It is observed that VVC performs worse than *Reconstruction* across all the tasks, which is potentially due to (1) the small spatial resolution (256x256) of input images that is not optimal for VVC, (2) its inferior rate-distortion performance compared to ELIC as reported in (He et al., 2022).

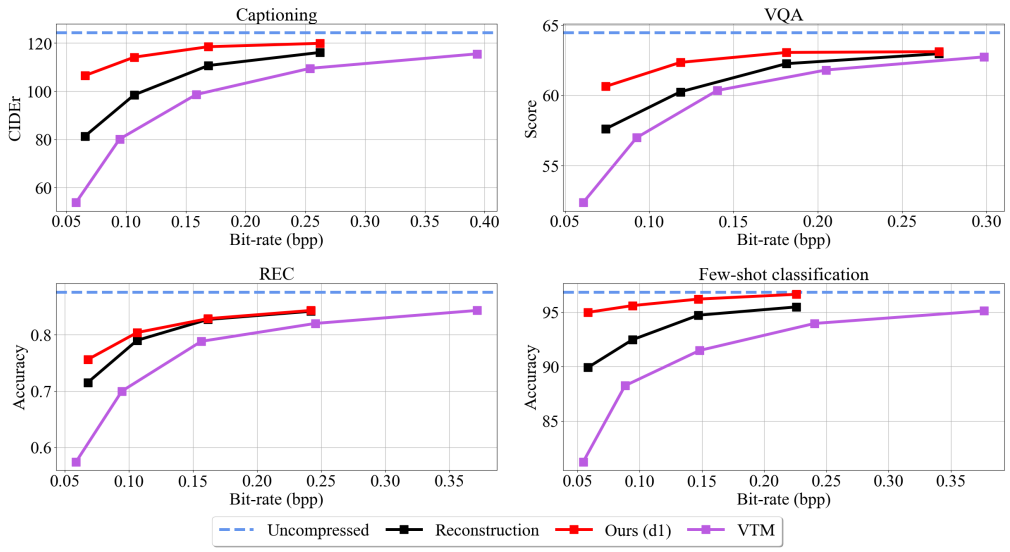


Figure 9: Rate-accuracy comparison using VTM on several tasks.

A.3 More Visualization

We present additional visualization results on four different evaluation tasks, including image captioning (Figure 12), visual question answering (VQA) (Figure 13), referring expression comprehension (REC) (Figure 14), and few-shot classification (Figure 15).

A.4 License of Assets Used

Table 5 summarizes the used assets in our work along with their license terms.

Table 5: List of assets used in the paper with their corresponding license.

Assets	Licenses
ImageNet Deng et al. (2009)	Custom license. Available at https://image-net.org/download.php
COCO Lin et al. (2014)	CC BY 4.0
SEED-Bench Li et al. (2023a)	Apache 2.0
LLaMA-Adapter Zhang et al. (2024a)	GPL-3.0
Honeybee Cha et al. (2024)	Source code: Apache 2.0 Pretrained weights: CC BY-NC 4.0
Shikra Chen et al. (2023b)	CC BY-NC 4.0
V2L-Tokenizer Zhu et al. (2024b)	No license provided. Code available at https://github.com/zh460045050/V2L-Tokenizer

A.5 Additional Performance Comparison

Figure 10 presents an additional performance comparison of our proposed method on the same task (POPE benchmark (Li et al., 2023c)) with two different MLLMs, Honeybee (Cha et al., 2024) and Shikra (Chen et al., 2023b).

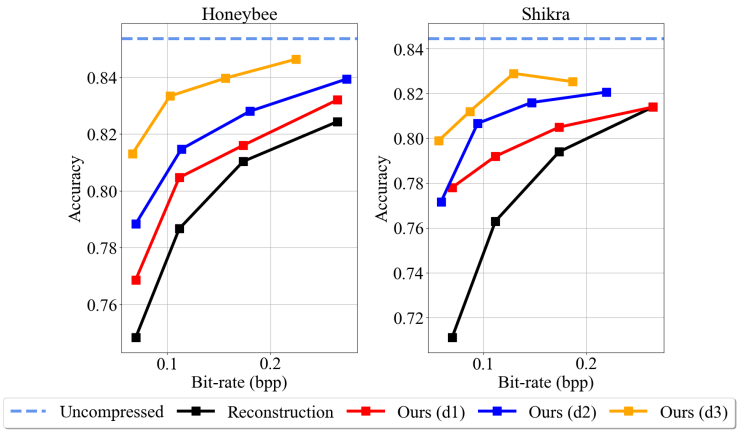


Figure 10: Performance comparison on POPE benchmark (Li et al., 2023c) with different MLLMs.

A.6 Comparison with Token Reduction Methods

Token reduction methods (Shi et al., 2024; Li et al., 2024) aim to reduce the number of visual tokens for lowering the inference computational cost of MLLMs. These methods differ fundamentally from our method as they do not consider the transmission of visual tokens in compressed form. In contrast, our work encodes and transmits the image in compressed form, where the compressed image latents are adapted to suit the downstream MLLM and image reconstruction tasks. The focus of our work is at maintaining downstream performance while reducing transmission cost and decoding inference cost. Figure 11 illustrates the different costs associated with each component in the system.

Ideally, one might propose generating visual tokens on the end device and using token reduction techniques as a compression method to reduce transmission bandwidth. However, this approach

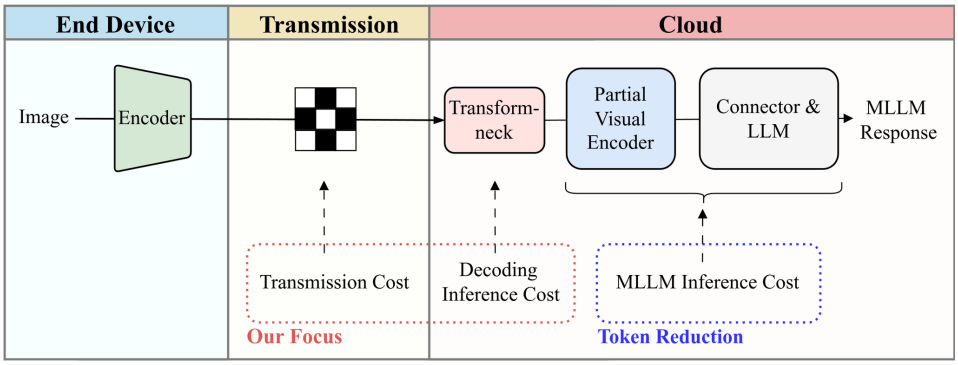


Figure 11: High-level architecture of our proposed method and different cost associated with each component.

would impose significant computational demands on the end device, making it impractical for coding for machines scenarios, where the primary goal is to offload heavy feature extraction computations to the cloud.

Notably, our method and the token reduction technique could potentially complement each other to develop a more efficient system. For instance, our approach allows to save transmission bandwidth and reduce cloud-side complexity by eliminating the need for image decoding, then token reduction techniques can further optimize complexity on the cloud.

Task: Captioning
Model: LLaMA-Adapter

Uncompressed | Reconstructed | Post-processed

BPP: 0.0725

Reconstruction: A microwave and a computer sitting on a desk.
Post-processing: A microwave and a refrigerator sitting on top of a table.
Ours (d1): A microwave and a toaster oven on a counter.

Uncompressed | Reconstructed | Post-processed

BPP: 0.0928

Reconstruction: Two cats are standing on the ground near a bench.
Post-processing: A dog and a cat are standing on a sidewalk.
Ours (d1): Two dogs are standing near a bicycle on a sidewalk.

Uncompressed | Reconstructed | Post-processed

BPP: 0.0910

Reconstruction: A blurry picture of a blender with a knife.
Post-processing: A close up of a blurry image of a bug.
Ours (d1): A close up of a knife cutting into a pizza.

Uncompressed | Reconstructed | Post-processed

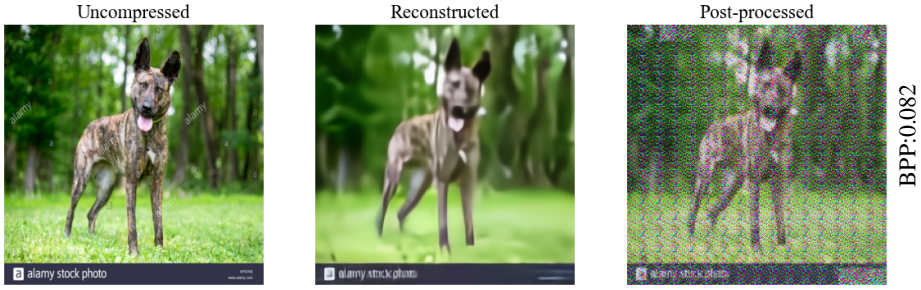
BPP: 0.0899

Reconstruction: A young boy in a red shirt and tie posing for a picture.
Post-processing: A young boy standing in front of a wall with a clock.
Ours (d1): A young boy in a tie and a white shirt.

Figure 12: Visualization examples of our proposed method in (d1), *Reconstruction*, and *Post-processing* on image captioning with LLaMA-Adapter.

Task: Visual question answering
Model: Honeybee

What is the dog doing in the image?
A. Standing still B. Chasing after something C. Lying down D. Jumping in the air

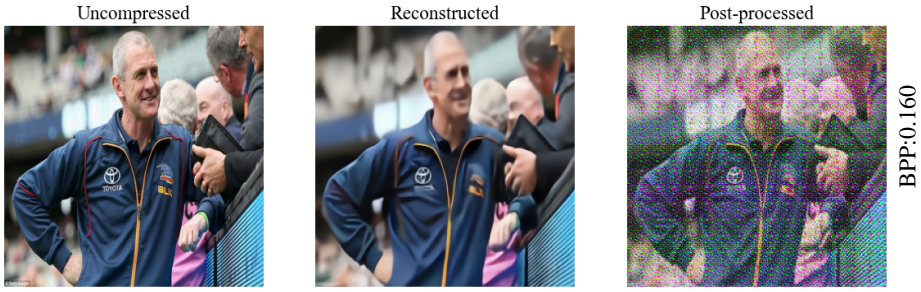


Reconstruction: B *Post-processing*: B Ours (d1): A GT: A

How many people are on the field in this image?
A. Four B. Nine C. Twelve D. Eleven

Reconstruction: D *Post-processing*: B Ours (d1): A GT: A

What is the person in the blue jacket holding?
A. A phone B. Nothing C. A wallet D. A clipboard



Reconstruction: D *Post-processing*: D Ours (d1): B GT: B

How many people are lighting candles in this image?
A. Two B. One C. Three D. Four

Reconstruction: A *Post-processing*: A Ours (d1): C GT: C

Figure 13: Visualization examples of our proposed method in (d1), *Reconstruction*, and *Post-processing* on VQA with Honeybee.

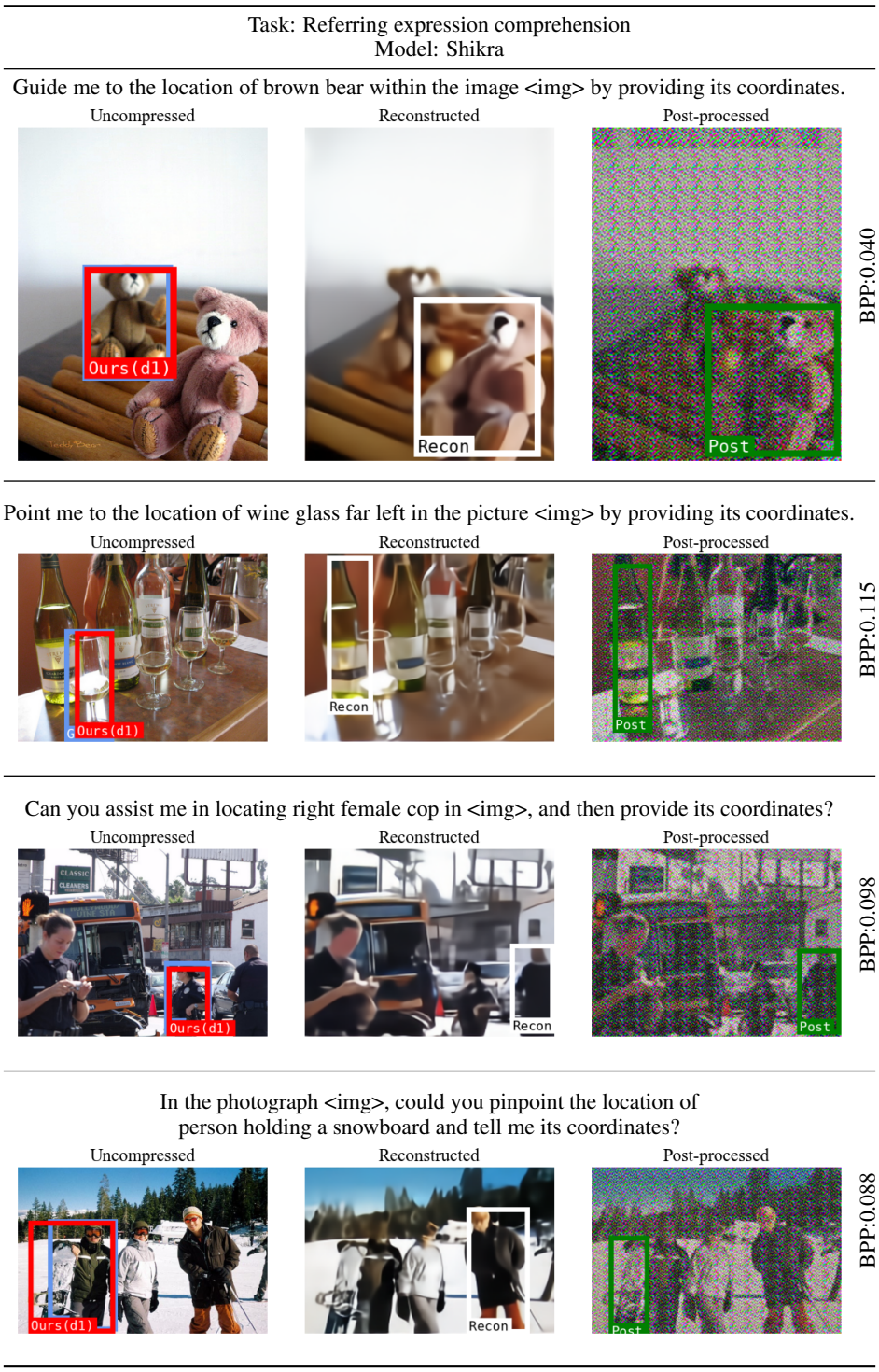


Figure 14: Visualization examples of our proposed method in (d1), *Reconstruction*, and *Post-processing* on REC with Shikra.

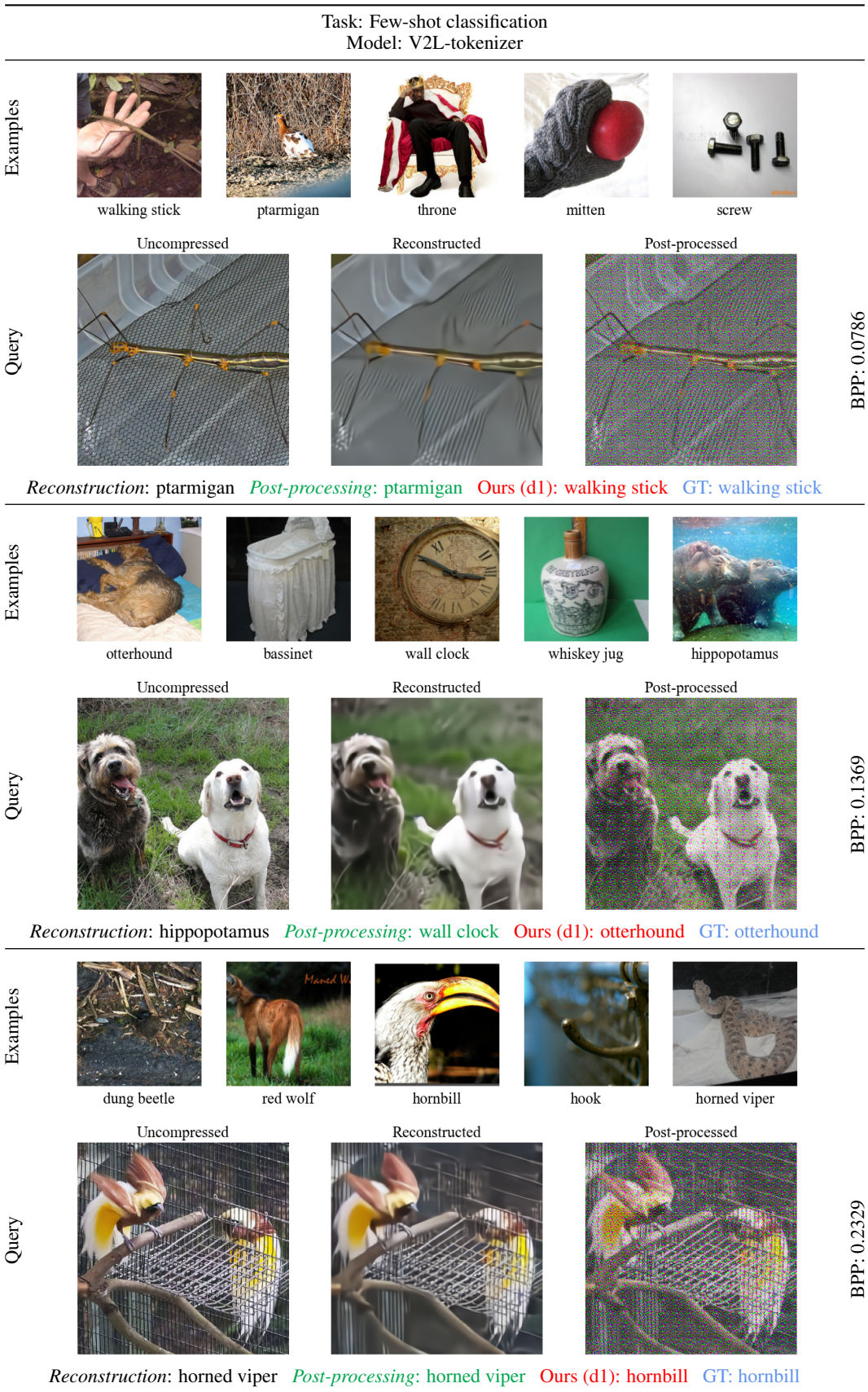


Figure 15: Visualization examples of our proposed method in (d1), *Reconstruction*, and *Post-processing* on few-shot classification with V2L-tokenizer.