Beyond Text: Frozen Large Language Models in Visual Signal Comprehension

Lei Zhu[1]   Fangyun Wei[2*]   Yanye Lu[1]

[1]Peking University   [2]Microsoft Research Asia

zhulei@stu.pku.edu.cn   fawe@microsoft.com   yanye.lu@pku.edu.cn

Abstract

In this work, we investigate the potential of a large language model (LLM) to directly comprehend visual signals without the necessity of fine-tuning on multi-modal datasets. The foundational concept of our method views an image as a linguistic entity, and translates it to a set of discrete words derived from the LLM's vocabulary. To achieve this, we present the Vision-to-Language Tokenizer, abbreviated as V2T Tokenizer, which transforms an image into a "foreign language" with the combined aid of an encoder-decoder, the LLM vocabulary, and a CLIP model. With this innovative image encoding, the LLM gains the ability not only for visual comprehension but also for image denoising and restoration in an auto-regressive fashion—crucially, without any fine-tuning. We undertake rigorous experiments to validate our method, encompassing understanding tasks like image recognition, image captioning, and visual question answering, as well as image denoising tasks like inpainting, outpainting, deblurring, and shift restoration. Code and models are available at https://github.com/zh460045050/V2L-Tokenizer.

1. Introduction

Significant advancements have been achieved in the field of natural language processing (NLP) through the deployment of large language models (LLMs), such as GPT [3, 30, 34, 35], PaLM [2, 6] and LLaMA [45, 46]. In pursuit of addressing intricate challenges necessitating the combination of text and visual understanding, scholars are broadening the capacities of the off-the-shelf LLMs. This enhancement involves the incorporation of additional visual processing components that facilitate the understanding of visual content [13, 23–25, 62] or the generation of images from text [41, 50, 59, 61]. Subsequently, these models undergo an extra re-training or fine-tuning using various multi-modal datasets to align the visual latent space with the language latent space. Nevertheless, the process generally requires a substantial amount of training resources.

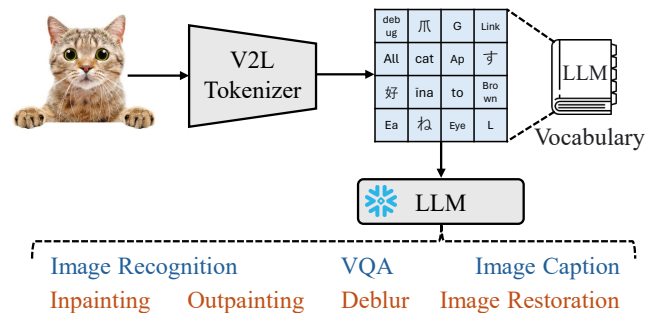As illustrated in Figure 1, this work aims to equip a

Figure 1. Illustration of our V2L Tokenizer (Vision-to-Language Tokenizer). The V2L Tokenizer translates an image into a collection of interpretable tokens derived from an LLM vocabulary. Subsequently, the frozen LLM can comprehend the visual signals and perform multi-modal understanding tasks (highlighted in Blue) and image denoising tasks (highlighted in Orange) without the necessity of fine-tuning.

large language model with the innate ability to comprehend visual signals, importantly, without the necessity of fine-tuning. In our approach, we view each image as a linguistic entity derived from a "foreign language", adapting it to suit the input requirements of a plain LLM. Consequently, this alignment occurs in the input (token) space rather than in the feature space, distinguishing our work from previous multi-modal methodologies [1, 23, 24, 62] that require fine-tuning for modality alignment. Thus, the fine-tuning or re-training process on multi-modal datasets is avoidable in our methodology. Our technique translates an image into a collection of discrete tokens that are within the vocabulary of the LLM. Once translated, these tokens can be fed into the LLM, enabling it to process and comprehend visual information, thereby facilitating a range of tasks involving both image understanding and denoising.

Translating an image into a set of tokens that a frozen LLM can understand is challenging. In this work, we introduce a tokenizer designed to map images (a non-linguistic modality) to the input (token) space of a frozen LLM. This tokenizer is termed the Vision-to-Language Tokenizer, or V2L Tokenizer in brief. Drawing inspiration from the triumphant advances of VQ-GAN [12], the V2L Tokenizer employs an encoder-quantizer-decoder structure. However,

its target is to translate visual information into the LLM's token space. This differs from its inspiration, which aims to learn an independent latent space solely for the purpose of image generation. Our V2L Tokenizer eschews the standard process of optimizing a randomly initialized quantizer codebook; instead, it leverages the pre-existing vocabulary of the LLM as its quantizer codebook throughout the training process. With the guidance of a quantization loss function, images are converted into a set of LLM tokens upon completion of the optimization process.

Typically, the vocabulary of an LLM consists of both full words and subword units due to the usage of language tokenizers such as BPE [42] and SentencePiece [19]. Without loss of generality, the breadth of this vocabulary influences its ability to encode images into LLM tokens—a larger vocabulary usually offers more powerful representation capabilities. In our approach, we expand the LLM's vocabulary by combining its lexical items to form bigrams or trigrams, which significantly augments the representation capacity when mapping an image into the LLM tokens. In addition to converting each image patch into a language token, our V2L tokenizer includes extracting global representations for the entire image. We accomplish this by utilizing a combination of subwords, bigrams, or trigrams from the expanded LLM vocabulary to encapsulate the image's comprehensive information.

In-context learning [3, 27, 28] has been shown to be highly beneficial for zero-shot inference in LLMs. This is accomplished by prefacing the instruction text with a number of domain-specific examples during the LLM inference. Our method eschews the necessity of LLM fine-tuning, instead employing in-context learning to guide the LLM in imitating the patterns presented in the given few-shot samples. This enables the model to better comprehend the "foreign language" (i.e., visual modality).

Experimentally, our work surpasses previous attempts [25, 54] in this novel scenario, where an LLM is able to comprehend visual signals without any fine-tuning or re-training, encompassing understanding tasks like image captioning and visual question answering, as well as image denoising tasks like inpainting, outpainting, deblurring, and image restoration.

2. Related Work

Image Quantization: The process of image quantization is designed to transform images into a series of discrete tokens derived from a codebook [12, 17, 20, 32, 33, 39, 48, 53, 55, 56]. VQ-VAE [48] stands as a notable work in the field. This method employs an encoder-decoder structure to quantize images into a collection of latent, discrete codes, which are then used to reconstruct the images. VQ-GAN [12] enhances the process of codebook learning by incorporating adversarial and perceptual losses, enabling the codebook

to capture more precise and finely detailed representations. Meanwhile, quantizing an image into a series of tokens enables image generation in an auto-regressive manner using GPT [3, 34, 35]. RQ-VAE [20] employs a residual quantization approach, where each image patch is represented by multiple codebook tokens, to more accurately mirror the original image features. DQ-VAE [17] further present tokens of variable length to encode images, resulting in more precise and efficient tokenization. Reg-VQ [56] aims to improve the utilization of the codebook and prevent its collapse by leveraging prior distribution regularization.

Large Language Models. Large language models (LLMs) [2, 3, 9, 38, 46, 58], especially those employing a Transformer-decoder architecture [2, 3, 46, 58], have made considerable progress in the domain of natural language processing. The process of developing an effective Large Language Model (LLM) generally unfolds in multiple phases, including initial pre-training [2, 3, 6, 15, 43], subsequent supervised fine-tuning [5, 13, 31, 57], the training of reward models [7, 10, 37], and the application of reinforcement learning using human feedback (RLHF) [14, 29–31, 46] to achieve alignment with instructions. The LLaMA [45] family has been at the forefront of offering open-source LLMs, providing both aligned and non-aligned versions in an array of scales [11, 18, 44–46, 51, 58]. For instance, the LLaMA 2 [46] presents models in the sizes of 7B, 13B, and 70B parameters.

Visual Signal Comprehension with LLMs: Despite the inherent capability for natural language understanding, LLMs can also act as decoders in various vision-language applications by employing a modality bridge module to align the visual with language features [1, 13, 21–24, 26, 49, 52, 60, 62]. For example, Flamingo [1] utilizes billions of image-text pairs to train gated cross-attention layers that facilitate the synchronization between a frozen vision encoder and a frozen LLM. In a similar vein, BLIP-2 [23] bridges the modality gap by introducing a lightweight Q-Former. This Q-Former is trained in two respective stages: one for representative learning and the other for generative learning. In addition, both MiniGPT-4 [62] and LLaVA [24] confirm that tuning a single linear layer on high-quality instruction data, is sufficient for feature alignment. While these methods yield satisfactory results for multi-modal understanding tasks, they lack the ability to generate visual content and necessitate the collection of additional image-text pairs to train the vision-language alignment modules.

Instead of performing multi-modal alignment in the feature space, several methods map images to the token (input) space of the LLMs by viewing images as "foreign languages" [25, 47, 54]. For instance, LQAE [25] trains a VQ-VAE tokenizer with a frozen LLM codebook to quantize an image into a set of language tokens. To enable an LLM to perform both image understanding and generation tasks,
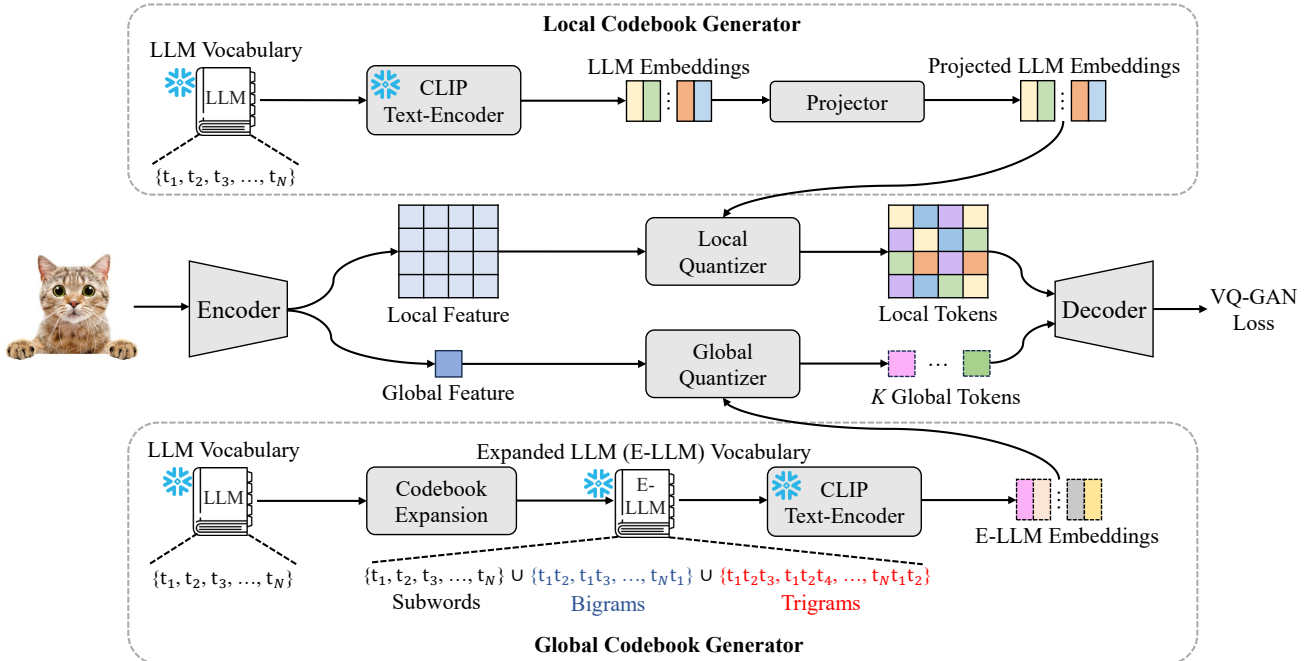
Figure 2. Overview of our Vision-to-Language Tokenizer (V2L Tokenizer). Figure 3 illustrates its integration with a frozen LLM.

SPAE [54] further enhances the quality of quantized image tokens derived from a frozen LLM codebook. It does so by incorporating a hierarchical quantization technique and semantic guidance provided by CLIP [36]. However, because of the substantial difference between visual features and language token embeddings, those methods struggle to assign semantic language tokens to images. This limitation hinders LLMs from fully understanding visual signals within a given context. In contrast to the aforementioned methods, our approach introduces image quantization within a shared multi-modal space, assigning semantically meaningful language tokens to a given image. Furthermore, we separate the image tokens into two categories: global tokens, which are used for image comprehension tasks, and local tokens, which are utilized for image generation tasks. This separation is accomplished through the use of two distinct types of quantizers along with two independent codebooks.

3. Method

3.1. Problem Formulation and Overview

We view images as a "foreign language". Given an LLM vocabulary $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$ containing N language tokens, we translate an image into K discrete tokens, each of which belongs to $\mathcal{T}$. This translation is accomplished by our V2L Tokenizer, as illustrated in Figure 2. In our implementation, an image is tokenized into K_g global tokens for understanding tasks, and K_l local tokens for denoising tasks, where $K = K_g + K_l$. Subsequently, as shown in Figure 3, we can perform a series of tasks such

as image classification, image caption, visual question answering, and image denoising. This is done by feeding the concatenation of task instructions, in-context learning samples, and either global or local tokens into a frozen LLM in an auto-regressive manner.

3.2. Vision-to-Language Tokenizer

Our Vision-to-Language Tokenizer (V2L Tokenizer) adopts an encoder-quantizer-decoder structure. In total, we employ two quantizers: a local quantizer and a global quantizer. Each of these is associated with an independent, frozen codebook derived from the LLM vocabulary. An image is then quantized into K_g global tokens and K_l local tokens, drawn from the global and local codebooks, respectively.

Global Codebook. An LLM vocabulary comprises a set of subwords generated by language tokenizers. These subword elements, in general, tend to have limited semantic significance. To enhance the semantic representation of entities within the LLM vocabulary $\mathcal{T}$ of size N, we introduce a vocabulary expansion technique. This technique entails creating bigrams and trigrams by combining two or three lexical items from $\mathcal{T}$. However, it is important to note that the resulting bigrams and trigrams may not necessarily convey meaningful semantics. For instance, they may include symbols like "#" and "!". Moreover, the generation of bigrams and trigrams leads to a vast number of possible combinations, which presents challenges in subsequent quantization processes.

To address this issue, we introduce a simple filter strategy. Specifically, using an image quantization dataset (such as ImageNet [8]) and the expanded LLM vocabulary, which

For each of the following input-out pairs, output is one of ['French bulldog, 'rock beauty'].

Input: Tokens( ), output: French bulldog.

Input: Tokens( ), output: rock beauty.

Input: Tokens( ), output:

(1) *N*-Way *K*-shot Classification

Generate a caption sentence based on words describing an image.

Input: Tokens( ), output: A man in a red shirt and a red hat is on a motorcycle on a hill side.

Input: Tokens( ), output: A woman wearing a hair net cutting a large sheet cake.

Input: Tokens( ), output:

(2) Image Caption

LLM → Prediction

Answer the question with a single word based on the condition.

Condition: Tokens( ),

Question: What is this person doing?
Answer: skiing.
Condition: Tokens( ),

Question: What does the truck on the left sell?
Answer:

(3) Visual Question Answering

Inpainting Outpainting Deblur Shift Rotation Masking

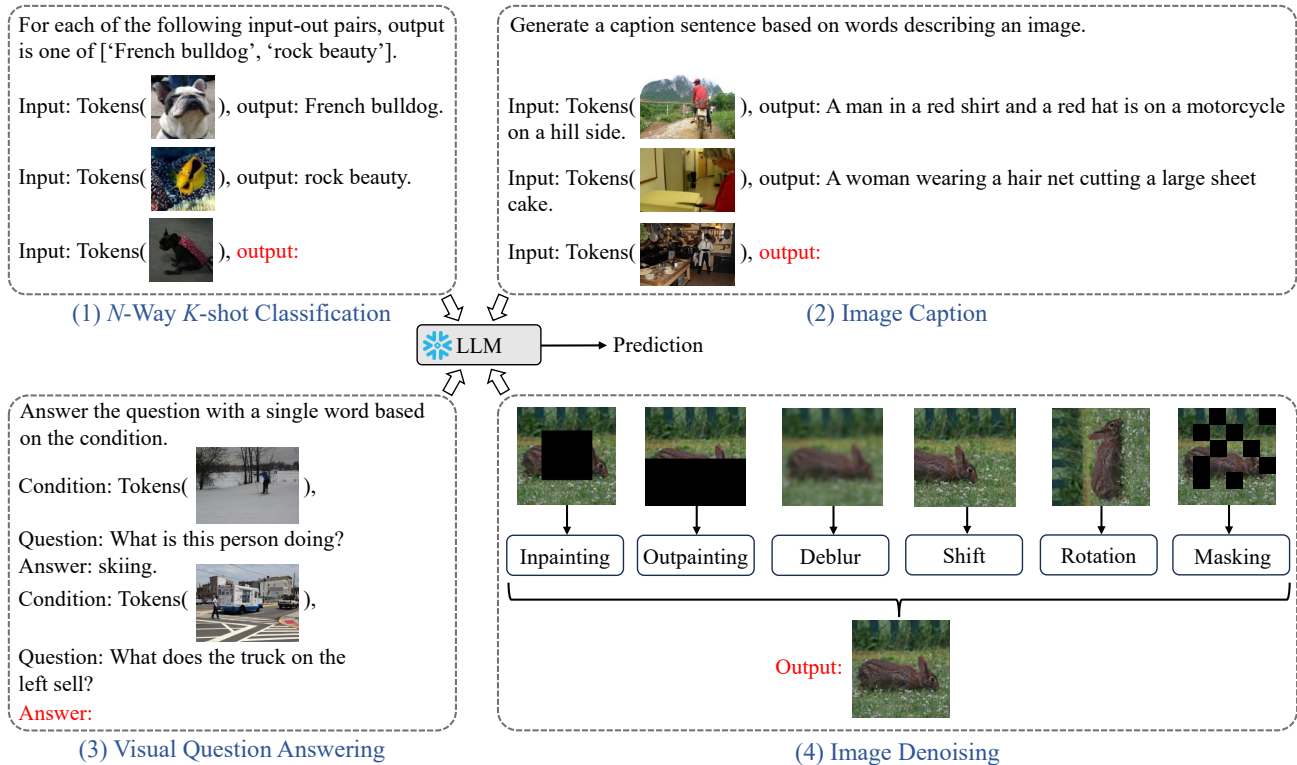Output:

(4) Image Denoising

Figure 3. Our V2L tokenizer enables a frozen LLM to perform a series of image understanding and denoising tasks.

includes all original subwords, bigrams, and trigrams, we compute the CLIP similarities [36] between each image in the dataset and every lexical item in the expanded LLM vocabulary. We then record the top-5 lexical items with the highest similarity scores for each image. Finally, we aggregate these top-5 lexical items from all images to form the final expanded LLM vocabulary, which serves as our global codebook.

Local Codebook. The objective of the local codebook is to use an item from this codebook to represent a part of an image (e.g., an image patch). We use the original LLM vocabulary as our local codebook.

Embeddings of Global and Local Codebooks. As illustrated in Figure 2, we project the global codebook (i.e., the expanded LLM vocabulary) and the local codebook (i.e., the LLM vocabulary) into embeddings through the CLIP-text-encoder [36]. The embeddings for the global and local codebooks are termed the LLM embeddings and the E-LLM embeddings, respectively. Additionally, we utilize a trainable projector, which is implemented as a linear layer, to further project the LLM embeddings for alignment with the visual space. The quantizers, which will be introduced later, further utilize the projected LLM embeddings (P-LLM embedding) and E-LLM embeddings to encode local and global information for an input image.

Encoder. Our encoder is composed of a trainable CNN encoder and a frozen CLIP-vision-encoder. The CNN encoder

is identical to the one used by VQ-GAN [12], but with modifications to the downsampling rate. We downsample the input image by a factor of 8. The CNN encoder aims to extract local information, while the CLIP-vision-encoder focuses on encoding global information. Refer to the supplementary materials for the details of the encoder.

Quantizers. We use $\boldsymbol{F} \in \mathbb{R}^{h \times w \times d_l}$ to denote the feature map encoded by the CNN encoder, where (h, w) is the spatial size. Similarly, $\boldsymbol{f} \in \mathbb{R}^{d_g}$ denotes the global feature encoded by the CLIP-vision-encoder, with d_g representing the dimension of $\boldsymbol{f}$. Let $\mathcal{E}_l$ denote the set of P-LLM embeddings of the LLM vocabulary $\mathcal{T}$, and $\mathcal{E}_g$ represent the set of E-LLM embeddings of the expanded LLM vocabulary $\mathcal{T}_E$, respectively.

As shown in Figure 2, the local quantizer operates by identifying the closest embedding in $\mathcal{E}_l$ for each element $\boldsymbol{F}_{(i,j)} \in \mathbb{R}^{d_l}$ within $\boldsymbol{F}$, where (i, j) specifies the spatial location ($1 \leq i \leq h$ and $1 \leq j \leq w$). The identification is based on Euclidean distance. This process yields a tokenized map $\widehat{\boldsymbol{F}}$ with the same size of $\boldsymbol{F}$. Each element $\widehat{\boldsymbol{F}}_{(i,j)} \in \mathcal{E}_l$ in $\widehat{\boldsymbol{F}}$ represents a P-LLM embedding associated with a language token belonging to $\mathcal{T}$. In total, there are $K_l = hw$ local tokens.

Similarly, the global quantizer functions by identifying the K_g closest embeddings in $\mathcal{E}_g$ for the global feature $\boldsymbol{f}$, based on their Euclidean distance. After quantization, $\boldsymbol{f}$ is represented by the K_g E-LLM embeddings $\widehat{\boldsymbol{f}} = \{\boldsymbol{e}_k \in$

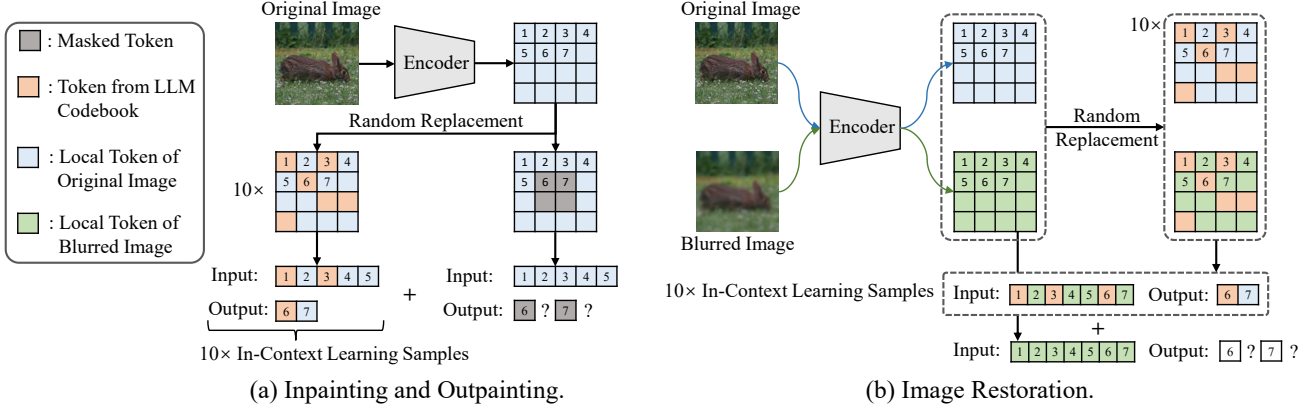(a) Inpainting and Outpainting.   (b) Image Restoration.

Figure 4. (a) We use inpainting as an example. Given an image, we first extract its local tokens $\mathcal{T}_l$. Following SPAE [54], we generate 10 copies for $\mathcal{T}_l$, termly $\{\mathcal{T}_l^s\}_{s=1}^{10}$. Each copy is a variation of $\mathcal{T}_l$ with tokens randomly replaced by those from the LLM codebook. The replacement ratios are set as [23%, 50%; 3%], where 3% denotes the incremental step. Next, an 8×8 mask (inpainting) or an 8×16 mask (outpainting) is applied to the center (inpainting) or the bottom (outpainting) of $\mathcal{T}_l$. The objective is to predict m masked tokens at a time using the first n tokens preceding them. The prompt is structured as follows: [*Learn a new language and predict m tokens following the examples.* $\{$*Input:* $\mathcal{T}_l^s[n]$, *output:* $\mathcal{T}_l^s[m]$.$\}_{s=1}^{10}$. *Input:* $\mathcal{T}_l[n]$, *output:*]. This prompt is then fed into the LLM, which sequentially predicts m tokens. Repeating this process enables us to predict all masked tokens. Finally, we organize these predictions along with the unmasked tokens and feed the complete token map into the decoder for image restoration. (b) We use deblurring as an example. Both shift and rotation restorations share similar principles. The prompt is structured as follows: [*Learn a new language and predict m tokens following the examples.* $\{$*Input:* $\overline{\mathcal{T}}_l^s[n+m]$, *output:* $\mathcal{T}_l^s[m]$.$\}_{s=1}^{10}$. *Input:* $\mathcal{T}_l[n+m]$, *output:*]. In this prompt, $\overline{\mathcal{T}}_l$ denotes the local tokens of the blurred image, $\overline{\mathcal{T}}_l^s$ indicates a variation of $\overline{\mathcal{T}}_l$ with tokens randomly replaced by those from the LLM codebook, and $\mathcal{T}_l^s$ represents the tokens of the original image, which undergo the same token replacement as $\overline{\mathcal{T}}_l^s$. By default, we set $n = 16$ and $m = 2$.

$\mathcal{E}_g\}_{k=1}^{K_g}$ associated with the corresponding language tokens $\{\boldsymbol{t}_k \in \mathcal{T}_E\}_{k=1}^{K_g}$. It should be noted that during the training of quantizers, both the LLM embeddings and the E-LLM embeddings remain frozen, as illustrated in Figure 2.

Decoder. The objective of the decoder is to reconstruct the original image by using the local embeddings $\widehat{\boldsymbol{F}}$ and the global embeddings $\widehat{\boldsymbol{f}}$ as inputs. Our decoder is built upon the one adopted by VQ-GAN [12], which utilizes a self-attention layer and a stack of transposed convolution layers to upsample $\widehat{\boldsymbol{F}}$ along the spatial dimension. The key distinction lies in the incorporation of $\widehat{\boldsymbol{f}}$: we inject the information of $\widehat{\boldsymbol{f}}$ into the decoding process through a cross-attention layer. In our implementation, this cross-attention layer is positioned following VQ-GAN's self-attention layer, where $\widehat{\boldsymbol{F}}$ serves as queries and $\widehat{\boldsymbol{f}}$ acts as keys. This modification does not affect the structure of the original decoder adopted by VQ-GAN. Consequently, the final output of the decoder is a tensor that matches the size of the input image.

Loss Function. As illustrated in Figure 2, we optimize only the encoder, the decoder, and the projector while freezing the LLM/E-LLM embeddings, the LLM/E-LLM vocabulary and the CLIP model. Following VQ-GAN, we define the objective function as:

$$\mathcal{L} = \mathcal{L}_{VQ} + \lambda_1 \mathcal{L}_{Perceptual} + \lambda_2 \mathcal{L}_{GAN},$$

where $\mathcal{L}_{VQ}$, $\mathcal{L}_{Perceptual}$ and $\mathcal{L}_{GAN}$ represent vector quantization loss, perceptual loss and GAN loss as introduced

by VQ-GAN, respectively; λ_1 and λ_2 denote the weights for the respective losses. We set $\lambda_1 = 1.0$ and $\lambda_2 = 0.1$. Refer to the original VQ-GAN [12] for more details on each type of loss.

3.3. Visual Signal Comprehension

We term the language tokens associated with $\widehat{\boldsymbol{f}}$ and $\widehat{\boldsymbol{F}}$ as global tokens (denoted as $\mathcal{T}_g = \{\boldsymbol{t}_k \in \mathcal{T}_E\}_{k=1}^{K_g}$) and local tokens (denoted as $\mathcal{T}_l = \{\boldsymbol{t}_k \in \mathcal{T}\}_{k=1}^{K_l}$), respectively, with the latter being after flattening. Note that $K_l = hw$, where (h, w) denote the spatial size of the feature map produced by the CNN encoder. Given an image, we first feed it into our V2L Tokenizer to generate its global tokens $\mathcal{T}_g$ and local tokens $\mathcal{T}_l$. Subsequently, we can design various prompts by combining task-specific introductions, in-context learning samples, as well as either global or local tokens, and feed the prompts into *a frozen LLM* to perform a series of understanding and generation tasks, as shown in Figure 3. We present the prompts for each task as follows.

N-Way K-Shot Image Classification. We use a 2-way K-shot classification as an example with the target of classifying images as either "French bulldog" or "Rock beauty". The prompt is structured as follows: [*For each of the following input-output pairs, output is one of ["French bulldog", "Rock beauty"]. {Samples}. Input:* $\mathcal{T}_g^{Test}$, *output:*], where $\mathcal{T}_g^{Test}$ denotes the global language tokens of the test image, and "{Samples}" signifies N-way K-shot samples. Each sample follows the format "Input: $\mathcal{T}_g$, output: L.", with $\mathcal{T}_g$

Method	#Tokens	Task Induction: N-way K-shot: #Repetitions:	✓ 2-1 0	✓ 2-1 0	✓ 2-3 0	✓ 2-5 0	✓ 2-1 1	✓ 2-1 3	✓ 2-1 5	Avg	✓ 5-1 0	✓ 5-1 0	✓ 5-3 0	✓ 5-5 0	✓ 5-1 1	✓ 5-1 3	✓ 5-1 5	Avg
Frozen [47]	-	-	1.7	33.7	66.0	66.0	63.0	65.0	63.7	51.3	0.9	14.5	34.7	33.8	33.8	33.3	32.8	26.3
LQAE [25]	256	GPT-3.5	1.5	35.2	68.2	69.8	68.5	68.7	65.9	54.0	1.0	15.7	35.9	36.5	31.9	36.4	45.9	29.0
SPAE [54]	5	GPT-3.5	5.3	77.2	84.4	86.0	79.4	77.2	77.1	69.5	-	-	-	-	-	-	-	-
SPAE [54]	5	PaLM-2 (340B)	32.2	84.0	88.5	88.4	85.1	83.6	82.4	77.7	23.6	64.2	68.0	69.9	63.4	62.0	60.2	58.8
Ours	5	LLaMA-2 (7B)	34.2	73.1	89.0	93.4	79.6	80.6	79.1	75.6	36.2	54.6	88.6	91.1	70.7	72.8	74.4	69.8
Ours	5	LLaMA-2 (13B)	44.4	77.9	91.9	94.4	81.5	82.8	82.0	79.3	**45.4**	69.6	89.9	91.3	75.8	75.7	77.2	75.0
Ours	5	LLaMA-2 (70B)	41.7	87.1	94.8	96.1	88.9	89.2	89.1	83.9	**45.4**	**81.5**	92.3	93.0	85.7	86.1	86.3	81.5
SPAE [54]	21	PaLM-2 (340B)	27.9	84.8	92.5	92.6	84.8	85.2	85.4	79.0	20.2	65.1	73.7	74.3	66.4	67.0	66.3	61.9
Ours	21	LLaMA-2 (7B)	36.5	76.3	91.2	95.3	84.0	84.4	83.7	78.8	37.1	44.8	91.8	94.0	73.9	82.2	85.3	72.7
Ours	21	LLaMA-2 (13B)	**48.7**	73.1	92.4	95.7	80.9	83.8	82.0	79.5	42.1	62.7	93.0	94.5	72.8	79.6	82.0	75.2
Ours	21	LLaMA-2 (70B)	46.5	**89.1**	**96.9**	**97.8**	**91.4**	**92.7**	**92.9**	**86.7**	45.0	79.7	**94.9**	**95.6**	**89.3**	**90.7**	**90.2**	**83.5**

Table 1. Few-shot Classification on 2-way and 5-way Mini-ImageNet benchmarks.

and L denoting the corresponding global tokens and the label (either "French bulldog" or "rock beauty") of each sample, respectively.

Image Caption. We structure the prompt as follows: [*Generate a caption sentence based on words describing an image.* {*Samples*}. *Input:* $\mathcal{T}_g^{Test}$, *output:*], where "{Samples}" denotes in-context learning samples. Each sample is formatted as "Input: $\mathcal{T}_g$, output: C", with $\mathcal{T}_g$ and C denoting the corresponding global tokens and the caption of each sample, respectively. The LLM takes this prompt as input and auto-regressively captions the test image with global tokens $\mathcal{T}_g^{Test}$, until it encounters the token ".".

Visual Question Answering. The prompt for VQA is designed as follows: [*Answer the question with a single word based on the condition.* {*Samples*}. *Condition:* $\mathcal{T}_g^{Test}$. *Question: Q. Answer:*], where $\mathcal{T}_g^{Test}$ denotes the global tokens of the test image, Q is the intended question, and "{Samples}" indicates in-context learning samples. Each sample has a format of "Condition: $\mathcal{T}_g$. Question: Q. Answer: A", with the triplet $(\mathcal{T}_g, Q, A)$ denoting the global tokens of one sample, the question related to this sample, and the ground-truth answer.

Image Denoising. We design several image denoising tasks following SPAE [54], including inpainting, outpainting, deblurring, shift restoration and rotation restoration. The prompts for those tasks are illustrated in Figure 4.

4. Experiments

4.1. Settings

We adopt LLaMA 2 [46] as our LLM, which has three versions with parameters of 7B, 13B, and 70B. Its vocabulary size is $32,000$. Our local codebook retains the original vocabulary from LLaMa 2. The size of the global codebook is $11,908$ after vocabulary expansion and filtering. The CLIP model used is the one with a ViT-L/14 backbone. Images are resized to a resolution of 128×128 pixels and are then

A dog is sitting in front of a computer.
A group of people in a kitchen.

A picture of a sign that says stop.
A bathroom with a bathtub and shower.

Q1: What food item is shown?
    Pizza   Burger
Q2: What country did this food originate from?
    Italy   Japan
Q3: What is the leafy substance?
    Basil   Lettuce

Figure 5. Visualizations for image caption (first row) and visual question answering (second row). Blue: ours. Orange: SPAE [54] (re-implementation).

processed by our V2L Tokenizer, which encodes them into a 16×16 token map. The training is conducted on the ImageNet-1K dataset over 100 epochs using 32 NVIDIA V100 GPUs. We use the Adam optimizer, starting with a learning rate of $5e^{-4}$, which undergoes a half-cycle cosine decay following a 5-epoch linear warm-up phase.

4.2. Image Comprehension

Few-Shot Classification. We conduct image comprehension experiments on 2-way and 5-way Mini-ImageNet benchmarks. All few-shot samples and test images are tokenized by our V2L Tokenizer into K_g global tokens. Then we structure the prompt as illustrated in Section 3.3 and Figure 3. This prompt is then input into the LLM for the purpose of predicting the category of the test image. It is important to note that the predictions are presented in text form. A prediction is considered correct only if all the generated tokens match the tokens of the actual category name.
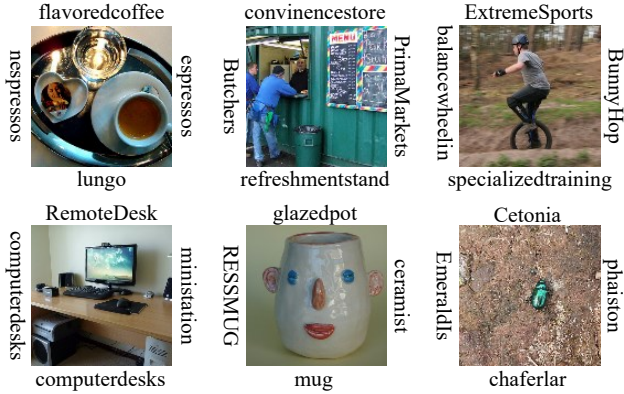
Table 1 shows the comparison between our approach em-

flavoredcoffee convinencestore ExtremeSports

nespressos espressos Butchers PrimaMarkets balancewheelin BunnyHop

lungo refreshmentstand specializedtraining

RemoteDesk glazedpot Cetonia

computerdesks ministation RESSMUG ceramist Emeralds phaiston

computerdesks mug chaferlar

Figure 6. Visualization for semantic interpretation.

Method	Codebook	#Tokens	CLIP↑	CLIP-R↑
SPAE [54]	PaLM-2	5	0.1868	0.7147
Ours	E-LLaMA-2	5	**0.2576**	**0.9165**
SPAE [54]	PaLM-2	21	0.1815	0.6901
Ours	E-LLaMA-2	21	**0.2427**	**0.8520**

Table 2. Semantic quality evaluatoin on ImageNet-1K val set. E-LLaMA-2: expanded LLaMa-2 vocabulary.

Method	Codebook	#Tokens	FID↓	LPIPS↓	PSNR↑
VQ-GAN [12]	Learnable	256	5.48	0.13	-
VQ-GAN [12]	PaLM-2	256	7.44	0.17	-
VQ-GAN* [12]	LLaMA-2	256	9.51	0.17	21.48
SPAE [54]	PaLM-2	341	9.49	0.17	-
SPAE [54]	PaLM-2	597	4.41	0.12	-
SPAE [54]	PaLM-2	1109	3.89	0.11	-
Ours	LLaMA2	256	3.41	**0.08**	**23.56**
Ours	Hybrid	277	**2.88**	**0.08**	23.25

Table 3. Reconstruction evaluation on ImageNet-1K val set. Hybrid: local tokens (256) and global tokens (21) are derived from the local codebook (LLaMA-2) and the global codebook (E-LLaMA-2), respectively. *: re-implementation.

In Table 2, we also quantitatively evaluate the semantic quality of our global tokens, and compare the semantic quality with SPAE [54] on ImageNet-1K validation set, using the CLIP score and the relative CLIP score (CLIP-R), which assess the degree of alignment between each image and its associated language tokens. We observe consistent improvements over SAPE, despite SAPE utilizing a larger vocabulary (SPAE's 65,000 versus our 11,908).

4.3. Image Reconstruction and Denoising

Reconstruction Evaluation. Our V2L Tokenizer encodes an image into a set of local tokens derived from an LLM vocabulary. These encoded tokens should capture the most meaningful information, enabling the decoder to reconstruct the original image and restore any degraded ("pollutional") images. In this study, we evaluate the reconstruction quality of our V2L Tokenizer using metrics including FID, LPIPS, and PSNR. As shown in Table 3, we compare our approach with SPAE [54] and VQ-GAN [12] on the ImageNet-1K validation set. In our approach, we explore two distinct setups: (1) employing the decoder from VQ-GAN without the involvement of global tokens; (2) utilizing the proposed decoder, which incorporates extra K_g global tokens for the decoding process (default configuration as discussed in Section 3.2). Our approach outperforms SPAE [54] across all metrics.

Image Denoising. We introduce the prompts used for inpainting, outpainting, deblurring, shift and rotation restorations, along with the process of restoring polluted images, as shown in Figure 4. In Table 4, we study two factors impacting the quality of these five in-context image denoising tasks: (1) the image tokenizer, which encodes an image into a set of tokens; (2) the LLM, which aims to predict the local tokens of the original images given the tokens of the polluted images, with the aid of in-context learning samples encoded by the tokenizer. The tokenizers used for comparison include VQ-GAN [12], LQAE [25], and SPAE [54]. We randomly select 5,000 images from the ImageNet-1K validation set to form our evaluation set. We use FID and LPIPS

ploying different LLaMa 2 model configurations, and prior works including LQAE [25], SPAE [54], and a baseline using a frozen language model for multimodal few-shot learning [47]. We examine various factors that could influence N-way K-shot classification, including: (1) the value of N; (2) the value of K; (3) task induction, defined as specifying the particular N-way categories for auto-regression; (4) the frequency of repetitions for each few-shot sample. We have two main observations: (1) Our model surpasses the previously best approach, SPAE [54], across all scenarios, despite using smaller LLMs (our 13B/70B LLaMa 2 versus SPAE's 340B PaLM-2) and a more compact vocabulary (our 11,908 versus SPAE's 65,000); (2) The performance of our model improves as the number of tokens used to represent the image increases. This can be attributed to the introduction of the vocabulary expansion, which generates a larger pool of semantically relevant token candidates.

Image Caption and Visual Question Answering. Following SPAE [54], we randomly select 10 image-caption pairs (or image-question-answer triplets) from COCO Caption [4] (or VQA [40]) training set to form the in-context learning samples in the image caption (or VQA) prompt. By default, we utilize 21 global tokens to represent an image. The visualization results are presented in Figure 5. Refer to supplementary materials for more results.

Semantic Interpretation. Figure 6 visualizes the top four global tokens with the highest similarity scores for a set of six images chosen at random. Our vocabulary expansion technique effectively increases the range of semantically pertinent token options (i.e. bigrams and trigrams). Extra results are available in the supplementary materials.

Tokenizer	LLM	Inpainting		Outpainting		Deblurring		Rotation		Shift	
		FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓
VQ-GAN* [12]	LLaMA-2 7B	16.44	0.1404	18.22	0.1571	13.79	0.1252	14.08	0.1285	13.91	0.1270
LQAE* [25]	LLaMA-2 7B	18.77	0.1736	19.61	0.1833	18.09	0.1711	18.18	0.1725	18.26	0.1722
SPAE* [54]	LLaMA-2 7B	14.89	**0.1211**	16.10	**0.1363**	15.89	0.1299	16.25	0.1318	16.55	0.1333
Ours	LLaMA-2 7B	**13.13**	0.1219	**15.28**	0.1442	**10.09**	**0.1033**	**10.64**	**0.1064**	**10.53**	**0.1058**
VQ-GAN* [12]	LLaMA-2 13B	15.56	0.1350	16.47	0.1449	14.78	0.1334	16.15	0.1417	15.60	0.1378
LQAE* [25]	LLaMA-2 13B	18.45	0.1720	18.78	0.1762	18.62	0.1740	19.04	0.1778	18.87	0.1770
SPAE* [54]	LLaMA-2 13B	13.89	0.1168	14.69	**0.1257**	16.46	0.1345	18.34	0.1436	17.71	0.1405
Ours	LLaMA-2 13B	**11.70**	**0.1134**	**12.56**	0.1275	**10.60**	**0.1085**	**11.36**	**0.1128**	**11.84**	**0.1176**
VQ-GAN* [12]	LLaMA-2 70B	14.08	0.1256	14.70	0.1358	14.30	0.1312	14.39	0.1313	14.35	0.1310
LQAE* [25]	LLaMA-2 70B	18.01	0.1692	18.54	0.1755	18.17	0.1713	18.16	0.1715	18.09	0.1713
SPAE* [54]	LLaMA-2 70B	12.79	0.1103	13.41	0.1191	18.08	0.1615	18.30	0.1619	18.19	0.1609
Ours	LLaMA-2 70B	**10.11**	**0.1021**	**10.73**	**0.1128**	**10.42**	**0.1058**	**10.48**	**0.1058**	**10.79**	**0.1093**

Table 4. Quantitative evaluation across five denoising restoration tasks. *: re-implementation.

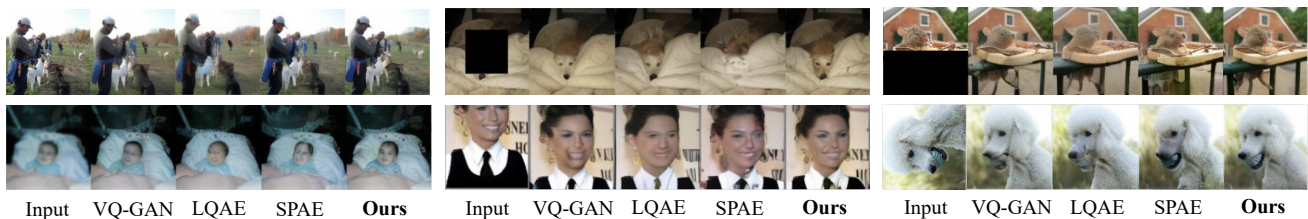Input VQ-GAN LQAE SPAE **Ours**   Input VQ-GAN LQAE SPAE **Ours**   Input VQ-GAN LQAE SPAE **Ours**

Figure 7. From left-to-right, top-to-bottom: visualizations for image reconstruction, inpainting, outpainting, deblurring, shift restoration and rotation restoration. We re-implement methods [12, 25, 54] using a vocabulary size of 32,000 and 256 local tokens for a fair comparison.

Input Prediction Input Prediction Input Prediction
Figure 8. Visualizations for masked image restoration.

scores as metrics. Our V2L Tokenizer outperforms others across the five tasks on almost all metrics. This achievement is attributed to the alignment of image features with the token space of the frozen LLM. We also show several qualitative results in Figure 7. More visualizations can be found in the supplementary materials.

Masked Image Restoration. Given an image from the ImageNet-1K validation set, we first extract its global and local tokens through our V2L Tokenizer. Subsequently, we apply random masking to 30% of these local tokens. To predict the masked tokens, we employ a LoRA-tuned [16] 7B LLaMa-2 model (details on tuning are provided in the supplementary materials). The next step involves integrating the predictions for the masked tokens with the unmasked tokens, which are then input into the decoder for image reconstruction. The qualitative results of this visual signal

restoration process are illustrated in Figure 8. For visualization purposes, the masked images ("input") presented are generated by combining the unmasked local tokens of the original image with the masked tokens which have been set to zero, before being processed through the decoder.

5. Conclusion

In this paper, we view images as a "foreign language", and introduce a V2L Tokenizer, which maps continuous visual signals to the token space of an LLM. Our method enables a frozen LLM to understand visual signals without the necessity for resource-intensive fine-tuning on multi-modal datasets. The V2T Tokenizer processes an image by generating both global and local tokens. The global tokens are crafted to capture essential semantic information with the aid of the proposed vocabulary expansion technique. This enables the execution of tasks like image recognition, image captioning and VQA. In contrast, local tokens are designed to extract detailed, patch-level features from images, facilitating image denoising tasks such as inpainting and deblurring. Extensive experiments validate the superiority of our approach over the prior attempts in this direction.

6. Acknowledgements

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 1, 2

[2] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 1, 2

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 2

[4] 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 preprint arXiv:1504.00325*, 2015. 7

[5] 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)*, 2023. 2

[6] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 1, 2

[7] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017. 2

[8] 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*, pages 248–255. Ieee, 2009. 3

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[10] Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023. 2

[11] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*, 2021. 2

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 1, 2, 4, 5, 7, 8

[13] 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. 1, 2

[14] Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022. 2

[15] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 2

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 8

[17] Mengqi Huang, Zhendong Mao, Zhuowei Chen, and Yongdong Zhang. Towards accurate image coding: Improved autoregressive image generation with dynamic vector quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22596–22605, 2023. 2

[18] Fangkai Jiao, Bosheng Ding, Tianze Luo, and Zhanfeng Mo. Panda llm: Training data and evaluation for open-sourced chinese instruction-following large language models. *arXiv preprint arXiv:2305.03025*, 2023. 2

[19] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018. 2

[20] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022. 2

[21] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 2

[22] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*, 2023.

[23] 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*, 2023. 1, 2

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 2

[25] Hao Liu, Wilson Yan, and Pieter Abbeel. Language quantized autoencoders: Towards unsupervised text-image alignment. *arXiv preprint arXiv:2302.00902*, 2023. 1, 2, 6, 7, 8

[26] Gen Luo, Yiyi Zhou, Tianhe Ren, Shengxin Chen, Xiaoshuai Sun, and Rongrong Ji. Cheap and quick: Efficient vision-language instruction tuning for large language models. *arXiv preprint arXiv:2305.15023*, 2023. 2

[27] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021. 2

[28] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022. 2

[29] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. 2

[30] OpenAI. Gpt-4 technical report, 2023. 1

[31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022. 2

[32] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021. 2

[33] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 2

[34] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1, 2

[35] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 2

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 4

[37] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. 2

[38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 2

[39] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2

[40] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. *Advances in neural information processing systems*, 28, 2015. 7

[41] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1

[42] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015. 2

[43] Saleh Soltan, Shankar Ananthakrishnan, Jack FitzGerald, Rahul Gupta, Wael Hamza, Haidar Khan, Charith Peris, Stephen Rawls, Andy Rosenbaum, Anna Rumshisky, et al. Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model. *arXiv preprint arXiv:2208.01448*, 2022. 2

[44] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: an instruction-following llama model (2023). *URL https://crfm. stanford. edu/2023/03/13/alpaca. html*, 1(2):3. 2

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 2

[46] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1, 2, 6

[47] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021. 2, 6, 7

[48] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[49] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023. 2

[50] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023. 1

[51] Honglin Xiong, Sheng Wang, Yitao Zhu, Zihao Zhao, Yuxiao Liu, Qian Wang, and Dinggang Shen. Doctorglm: Fine-tuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097*, 2023. 2

[52] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 2

[53] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 2

[54] Lijun Yu, Yong Cheng, Zhiruo Wang, Vivek Kumar, Wolfgang Macherey, Yanping Huang, David A Ross, Irfan Essa, Yonatan Bisk, Ming-Hsuan Yang, et al. Spae: Semantic pyramid autoencoder for multimodal generation with frozen llms. *arXiv preprint arXiv:2306.17842*, 2023. 2, 3, 5, 6, 7, 8

[55] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 2

[56] Jiahui Zhang, Fangneng Zhan, Christian Theobalt, and Shijian Lu. Regularized vector quantization for tokenized image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18467–18476, 2023. 2

[57] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 2

[58] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 2

[59] Tianjun Zhang, Yi Zhang, Vibhav Vineet, Neel Joshi, and Xin Wang. Controllable text-to-image generation with gpt-4. *arXiv preprint arXiv:2305.18583*, 2023. 1

[60] Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*, 2023. 2

[61] Shanshan Zhong, Zhongzhan Huang, Weushao Wen, Jinghui Qin, and Liang Lin. Sur-adapter: Enhancing text-to-image pre-trained diffusion models with large language models. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 567–578, 2023. 1

[62] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2