FROM PIXELS TO TOKENS: BYTE-PAIR ENCODING ON QUANTIZED VISUAL MODALITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models have made significant strides in integrating visual and textual information, yet they often struggle with effectively aligning these modalities. We introduce a novel image tokenizer that bridges this gap by applying the principle of Byte-Pair Encoding (BPE) to visual data. Unlike conventional approaches that rely on separate visual encoders, our method directly incorporates structural prior information into image tokens, mirroring the successful tokenization strategies used in text-only Large Language Models. This innovative approach enables Transformer models to more effectively learn and reason across modalities. Through theoretical analysis and extensive experiments, we demonstrate that our BPE Image Tokenizer significantly enhances MLLMs' multimodal understanding capabilities, even with limited training data. Our method not only improves performance across various benchmarks but also shows promising scalability, potentially paving the way for more efficient and capable multimodal foundation models.[1]

1 INTRODUCTION

The development of Multimodal Large Language Models (MLLMs) has made significant progress (Yin et al., 2023; Team et al., 2023; Liu et al., 2024b). However, these multimodal foundation models often model different modalities separately, incorporating many modality-specific designs such as specialized encoders and decoders (Liu et al., 2024b; Zhang et al., 2024; Jin et al., 2023). While this approach allows training data to align well with these modality-specific designs, it often struggles to achieve a unified understanding of multimodal information (Team, 2024). The primary reason for this limitation could be that while encoders of other modalities can learn rich information, without the assistance of the corresponding decoders, LLMs cannot fully comprehend the complex patterns contained within the embeddings provided by the encoder. In other words, LLMs need to learn to interpret the token embeddings again, which is the job of the decoders of other modalities, leading to difficulties in aligning with these modalities (Baltrušaitis et al., 2018).

Recent research has begun to explore unified token-based representations for MLLMs (Team, 2024; Lu et al., 2024; Zheng et al., 2024), attempting to achieve better capabilities for multimodal information processing by moving away from modality-specific designs. This approach offers two main advantages: first, it can achieve unified information understanding, and second, it enables unified generation that can be decoded into different modalities. However, these methods simply convert various modal information into unified representations and then directly feed them into Transformer models (Vaswani, 2017), hoping to learn the data just with massive computing resources. Although this approach can be effective to some extent, it consumes enormous computing resources and time while failing to achieve significant improvements in performance.

In comparison to the widely adopted training paradigm of text-only LLMs (Touvron et al., 2023; Achiam et al., 2023), we observe that current token-based MLLMs often overlook a crucial component: the tokenizer that explicitly merges data sources. In text-only LLMs, the most widely adopted tokenizer is the Byte-Pair Encoding (BPE) tokenizer (Sennrich, 2015; Radford et al., 2019), which learns to merge characters into tokens of varying lengths based on the word frequency in the training corpus before providing them to the Transformer for learning. While many believe this approach is

[1]The code is anonymously available at: https://anonymous.4open.science/r/BPE-Image-Tokenizer/

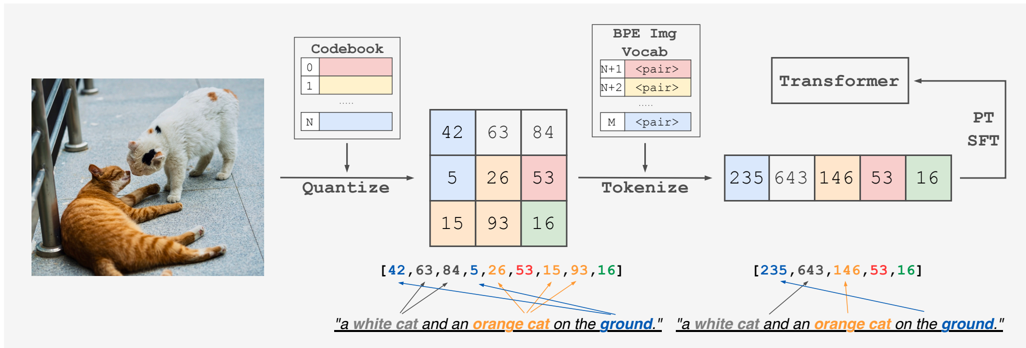


Figure 1: Illustration of our BPE Image Tokenizer. The overall process begins with quantizing the image into initial token IDs. The BPE Image Tokenizer then combines these tokens based on learned patterns, similar to text tokenizers. This combination results in tokens that inherently contain more semantic information. The final tokenized sequence thus incorporates structural prior information from the image, enabling the Transformer model to deeper comprehend the alignment between visual and textual information during training. This approach facilitates more effective integration of visual data into MLLMs, enhancing their multimodal understanding capabilities.

solely for conserving computational resources, some recent theoretical and experimental analyses suggest that using merged tokens plays a vital role in the Transformer's learning of sequential data (Rajaraman et al., 2024; Makkuva et al., 2024; Merrill et al., 2024). In fact, if raw sequence data is provided directly, Transformer models may even fail to learn how to predict entirely.

In this paper, we reveal that Transformer models cannot effectively learn certain types of two-dimensional sequence data. However, when we apply operations similar to BPE tokenizers to the two-dimensional data, *i.e.*, merging tokens based on the frequency of training data, the learning effectiveness can improve significantly. We believe the benefit comes from the addition of necessary structural prior information to the tokens during the merging process, making it easier for Transformer models to be aware of important relational information in the data during learning. Our theoretical analysis proves that tokenizing sequence data on certain types of two-dimensional data can indeed achieve smaller losses, which we further validate experimentally.

Based on the insights provided by our theoretical analysis and experimental validation, we propose a new learning paradigm for MLLMs. By tokenizing the unified representation of multimodal data using a novel BPE Image Tokenizer, we enable Transformer models to better understand image data, allowing MLLMs built from text-only LLMs to have good multimodal capabilities. As illustrated in Figure 1, after quantizing the image into token IDs, the BPE Image Tokenizer further combines these token IDs according to its learned patterns. Similar to text tokenizers, our BPE Image Tokenizer ensures that the tokens being fed to the language model decoder inherently contain more semantically meaningful information in the statistical sense. Intuitively, the tokenized IDs, which have incorporated structural prior information from the image, could provide the Transformer model with better comprehension of the specific alignment between text and image in the training data. Preliminary results further validate the effectiveness of explicit tokenization for multimodal data. We hope that the new paradigm for learning MLLMs proposed in this paper will provide better guidance for subsequent scaling-up efforts, leading to the building of more powerful MLLMs.

In summary, we make the following key contributions:

- We are the first to propose a new MLLM learning paradigm that explicitly tokenizes multimodal data like text-only LLMs, centered around our novel BPE Image Tokenizer.
- We theoretically analyze why this learning paradigm can bring benefits and further provide corresponding experimental validation.
- We design an algorithm for training the BPE Image Tokenizer and train an MLLM with this tokenizer. The performance evaluation further validates the capability enhancements this learning paradigm brings to MLLMs.

2 Notations and Formulation

Before delving into the theoretical analysis of our proposed paradigm, we introduce key notations and concepts used throughout this paper.

Image Representation and Quantization. We represent an image as a set of patches $\mathcal{X} = \{X_{ij}\}_{1 \le i,j \le m}$, where m is the number of patches per row/column, and X_{ij} denotes the patch at position (i, j). We employ Vector Quantization (VQ) to discretize these patches, using a codebook $\mathcal{C}$ with size $C = |\mathcal{C}|$, and a quantization function $\text{VQ} : \mathbb{R}^d \to \mathcal{C}$.

BPE Image Tokenizer. Our proposed BPE Image Tokenizer converts a quantized image into a sequence of token IDs, defined as $\text{enc} : \mathcal{X} \to \mathcal{T}^*$, where $\mathcal{T}$ is the set of tokens and $\mathcal{T}^*$ is the set of all finite sequences of tokens. A special case of encoding is flattening, $\text{flat} : \mathcal{X} \to \mathcal{C}^{m^2}$, which arranges image patches into a sequence row by row.

Unigram Model. For a unigram model $Q \in \mathcal{Q}_{1\text{-gram}}$, given a token sequence $\mathbf{t} = (t_1, ..., t_{|\mathbf{t}|})$, the probability is defined as:

$$Q(\mathbf{t}) = Q_{\#}(|\mathbf{t}|) \prod_{r=1}^{|\mathbf{t}|} Q_{\text{tok}}(t_r), \tag{1}$$

where $Q_{\#}$ is a distribution over $\mathbb{N}$ representing the probability of the sequence length, and Q_{tok} is a distribution over the dictionary of tokens $\mathcal{T}$.

Multimodal Large Language Model (MLLM). We define an MLLM as a probabilistic model Q over token sequences, capable of processing both text and image data as follows:

1. Input Processing: $\mathbf{t}_{\text{text}} = \text{tokenize}(\mathbf{x}_{\text{text}})$; $\mathbf{t}_{\text{image}} = \text{enc}(\text{VQ}(\mathbf{x}_{\text{image}}))$.
2. Sequence Modeling: $P(t_i | t_1, ..., t_{i-1}) = Q(t_i | \mathbf{t}_{<i})$.
3. Generation: $y_i \sim P(t_i | y_1, ..., y_{i-1}, \mathbf{t})$

The training objective for an MLLM is to minimize the cross-entropy loss:

$$\mathcal{L} = -\mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim D} \left[\sum_{i=1}^{|\mathbf{y}|} \log P(y_i | y_{<i}, \mathbf{t}) \right], \tag{2}$$

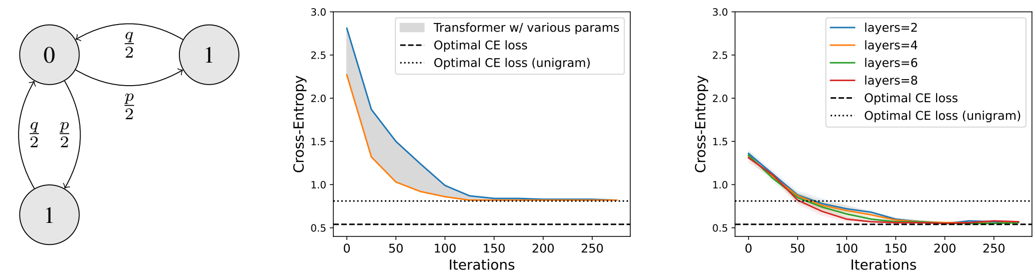
where D is a dataset of input-output pairs $(\mathbf{x}, \mathbf{y})$. The goal is to find the optimal parameters θ^* that minimize this loss: $\theta^* = \arg\min_\theta \mathcal{L}(\theta)$.

3 Theoretical Analysis

Previous studies have demonstrated that Transformers struggle to effectively learn certain one-dimensional sequence data (Rajaraman et al., 2024; Makkuva et al., 2024). In this section, we extend this concept to two-dimensional image data, considering a simplified model where the image data-generating distribution follows a 2D k^{th}-order Markov process. This simplified model is defined as follows:

Definition 1 (2D k^{th}-order Markov process)**.** *For each variable $X_{i,j}$, with probability $\frac{1}{2}$, it depends only on $X_{i-k,j}$, i.e., $Pr(X_{i,j} = 1 | X_{i-k,j} = 0) = p$ and $Pr(X_{i,j} = 1 | X_{i-k,j} = 1) = 1 - q$; With probability $\frac{1}{2}$, it depends only on $X_{i,j-k}$, i.e., $Pr(X_{i,j} = 1 | X_{i,j-k} = 0) = p$ and $Pr(X_{i,j} = 1 | X_{i,j-k} = 1) = 1 - q$. Figure 2(a) illustrates this process.*

This simplification is intuitive, as pixels in an image often exhibit conditional dependencies with other pixels at specific horizontal and vertical distances. Consequently, real-world image data can be viewed as a composite of multiple such simplified processes. When attempting to learn such sequence data directly using a Transformer model, an interesting phenomenon occurs: the Transformer fails to surpass the performance of the stationary unigram model, regardless of various hyperparameter choices. As shown in Figure 2(b), the Transformer fails to improve the cross-entropy loss beyond that of the optimal unigram model on the training data (indicated by the dotted line). However, as



(a) 2D k^{th}-order Markov process. Sequence data is transformed based on horizontal or vertical k^{th}-order Markov conditional probabilities.

(b) Without tokenizer, Transformer fails to learn 2D k^{th}-order Markov sequence data, only achieving the performance of a unigram model (dotted line). The gray area represents Transformer under various parameters

(c) With BPE tokenizer, even Transformer models with only a few layers (less parameters) can easily learn the 2D k^{th}-order Markov sequence data, achieving optimal cross-entropy loss (dashed line).

Figure 2: Definition of 2D k^{th}-order Markov sequence data, and the performance of Transformer in learning such sequence data with or without tokenizer. For details of the hyperparameters used in the experiments, please refer to section B.

shown in Figure 2(c), when encoding the sequences in each direction separately using BPE Tokenizer trained on this distribution, even a minimal Transformer model with only a few layers can readily achieve optimal loss. This finding suggests that BPE Tokenizer could play a crucial role in the Transformer's learning process for the defined two-dimensional sequence data.

To formally analyze this phenomenon in a more general setting, we first let m be the number of patches per row in an square image, and let X_{ij} denote the patch at position (i, j) for $1 \leq i, j \leq m$. An image can be represented by the set of m^2 patches $\{X_{ij}\}_{1 \leq i,j \leq m}$, which we abbreviate as $\boldsymbol{X}$ for simplicity. Each patch can be encoded into an index with a VQ-GAN model's codebook, denoted by $\mathcal{C}$, with $C = |\mathcal{C}|$ representing the size of the codebook.

We then consider two scenarios for image generation:

Scenario 1. *Every column in the image $\boldsymbol{X}$ is generated independently by an ergodic Markov process supported on $\mathcal{C}$, proceeding from top to bottom, with the same transition kernel $P(\cdot \mid \cdot)$ and stationary distribution π.*

Scenario 2. *Every row in the image $\boldsymbol{X}$ is generated independently by an ergodic Markov process supported on $\mathcal{C}$, proceeding from left to right, with the same transition kernel $P(\cdot \mid \cdot)$ and stationary distribution π.*

Given a model Q on tokens and an encoding function $\mathrm{enc}(\cdot)$ that encodes an image into a sequence of tokens, we define the cross entropy loss as

$$\mathcal{L}_m(Q \circ \mathrm{enc}) = -\frac{1}{m^2}\mathbb{E}[\log(Q(\mathrm{enc}(\boldsymbol{X})))],$$

where the subscript m indicates the dependence on size m. We also introduce the trivial encoding function $\mathrm{flat}(\cdot)$, which merely flattens an image into a sequence of patches arranged row by row from left to right:

$$\mathrm{flat}(\boldsymbol{X}) = (X_{11}, X_{12}, ...X_{1m}, X_{21}, X_{22}, ..., X_{2m}, ..., X_{m1}, X_{m2}, ..., X_{mm}).$$

Using this notation, $\mathcal{L}_m(Q \circ \mathrm{flat})$ represents the cross entropy loss for model Q along with the encoding function $\mathrm{flat}(\cdot)$. For simplicity, we abbreviate it as $\mathcal{L}_m(Q)$ throughout this paper. We define $H(p)$ as the Shannon entropy for any discrete distribution p:

$$H(p) = -\sum_{y \in \mathrm{supp}(p)} p(y) \log p(y).$$

Now, we present our main theoretical results:

Proposition 1. *For data generating processes described in either Scenario 1 or Scenario 2, as $m \to \infty$, the optimal cross-entropy loss among unigram model family $\mathcal{Q}_{\text{1-gram}}$ satisfies*

$$\liminf_{m\to\infty} \min_{Q\in\mathcal{Q}_{\text{1-gram}}} \mathcal{L}_m(Q) \geq H(\pi) = \sum_{a\in\mathcal{C}} \pi(a)\log(\pi(a)). \tag{3}$$

In contrast, the optimal unconstrained cross entropy loss satisfies

$$\lim_{m\to\infty} \min_{Q} \mathcal{L}_m(Q) = H_\infty \triangleq -\sum_{a\in\mathcal{C}}\sum_{a'\in\mathcal{C}} \pi(a)P(a' \mid a)\log\left(P(a' \mid a)\right). \tag{4}$$

Here we provide a sketch of the proof of Proposition 1. For the full proof, please see Appendix A.1.

Sketch of Proof. For the optimal cross-entropy loss among $\mathcal{Q}_{\text{1-gram}}$, observe that under both scenarios, the marginal distribution of each patch is precisely the stationary distribution π. When applying a unigram model to $\text{flat}(\boldsymbol{X})$ and leveraging its inherent independence, the optimal loss closely approximates $H(\pi)$, with a discrepancy that vanishes as $m \to \infty$. Regarding the optimal unconstrained cross-entropy loss, note that the best model is simply the generating model of $\text{flat}(\boldsymbol{X})$, with its cross-entropy loss converging to H_∞ as $m \to \infty$. $\square$

Remark 1. *Consider the 2D k^{th}-order Markov process in Figure 2(a). Let $\frac{p}{2} = \frac{q}{2} = \frac{1-\delta}{2}$, which means switching between 0 and 1 with equal probability $p = q = 1-\delta$. For this process, $\lim_{m\to\infty} \frac{1}{m}H(P) = \delta\log(\frac{1}{\delta}) + (1-\delta)\log(\frac{1}{1-\delta})$. However, the unigram stationary distribution is $\pi = \{\frac{1}{2}, \frac{1}{2}\}$, and so $H(\pi) = \log(2)$. The ratio $\lim_{m\to\infty} \frac{H(\pi)}{\frac{1}{m}H(P)}$ approaches ∞ as $\delta \to 0$. This example shows that the gap between $H(\pi)$ and H_∞ could be arbitrarily large.*

The implications of Proposition 1 and Remark 1 are profound for understanding the limitations of simple models in capturing the structure of two-dimensional image data. This result shows a significant performance gap between unigram models and the optimal unconstrained model for such data. This gap, represented by the difference between $H(\pi)$ and H_∞, can be arbitrarily large. This finding underscores a critical limitation of simple unigram models, which Transformers could default to when processing raw sequences, according to our earlier experimental results. These models are inherently restricted in their ability to capture the true structure of image data, suggesting that more sophisticated tokenization methods are necessary for effectively processing two-dimensional image data with Transformer models.

Proposition 2. *For data generating processes described in either Scenario 1 or Scenario 2, assume that $\delta \triangleq \min_{a,a'\in\mathcal{C}} P(a'|a) > 0$. Then there exists a tokenizer with a dictionary containing at most D tokens, along with an encoding function $\text{enc}(\cdot)$ applied to $\boldsymbol{X}$, such that*

$$\limsup_{m\to\infty} \min_{Q\in\mathcal{Q}_{\text{1-gram}}} \mathcal{L}_m(Q \circ \text{enc}(\cdot)) \leq \frac{1}{1-\varepsilon} H_\infty, \tag{5}$$

where $\varepsilon = \log(1/\delta)/(0.99\log(D))$ and $D \in \mathbb{N}$ is an arbitrary constant that is sufficiently large.

Here we provide a sketch of the proof of Proposition 2. For the full proof, please see Appendix A.1.

Sketch of Proof. The proof leverages the fact that in both scenarios, the generated image is composed of m parallel Markov sequences, each of length m, sharing the same transition kernel and stationary distribution. We first apply the result from the one-dimensional case (Lemma A.1) to each column (or row) separately. Then, we construct a tokenizer for the entire image by concatenating the tokens from each column (or row). By carefully bounding the cross-entropy loss of this constructed tokenizer, we show that it satisfies the desired inequality. The key step is to handle the additional complexity introduced by the two-dimensional structure while maintaining the bound derived from the one-dimensional case. $\square$

Proposition 2 offers valuable insights into the potential benefits of using a tokenizer for image data. This result demonstrates that with an appropriate tokenizer, we can achieve a loss arbitrarily close to the optimal unconstrained loss H_∞, significantly outperforming the unigram model bound of $H(\pi)$. These theoretical results motivate the design of our BPE Image Tokenizer. In the following sections, we will describe the implementation of our algorithm and present corresponding experimental results.

Algorithm 4.1 BPE Image Tokenizer training procedure.

```
Input v_0, m, D.            ▷ v_0: initial vocab size, m: new vocab size, D: training data
v ← v_0                     ▷ v: current vocab size
A ← zeros(v × v)            ▷ A: adjacency matrix
V ← ∅                       ▷ V: extended vocabulary
for i ← 1 to m do
    A ← UpdateMatrix(D)
    (p, f) ← MaxFreqPair(A)  ▷ p: best pair, f: frequency
    if f = 0 then break
    end if
    V ← V ∪ {(p, v)}
    D' ← ∅
    for each d ∈ D do
        d' ← Replace p with v in d
        D' ← D' ∪ {d'}
    end for
    D ← D'
    v ← v + 1                ▷ set next id for new token
end for
return V
```

4 Preliminary Implementation

To validate the effectiveness of our proposed BPE Image Tokenizer and the training paradigm in enhancing models' multimodal understanding capabilities, we implemented the entire procedure and conducted a series of experiments. Given the scope of this paper, we emphasize that this implementation for training an MLLM is preliminary and does not involve extensive scaling in terms of dataset size or computational resources to directly compare with state-of-the-art foundation models.

4.1 BPE Image Tokenizer Training

Dataset Preparation: We constructed a diverse image dataset comprising 2.78 million images from ImageNet (Deng et al., 2009), CC (Sharma et al., 2018), LAION (Schuhmann et al., 2022), and SBU (Ordonez et al., 2011). We applied a filtering mechanism similar to LLaVA (Liu et al., 2024b), selecting images based on their class labels or noun-phrase statistics derived from captions to keep the concept diversity.

Image Preprocessing: We used a pretrained VQ-GAN model released by Chameleon (Team, 2024) to preprocess the images. This VQ-GAN model uses a codebook size of 8192 and quantizes images of any size into a 1024-dimensional ID tensor.

Tokenizer Training Algorithm: We developed the BPE image tokenizer training algorithm, inspired by the text BPE algorithm. The main procedure of this algorithm is presented in Algorithm 4.1, with supporting functions detailed in Algorithms B.1 and B.2 (see Appendix B). Key designs of our algorithm include:

- Use of an adjacency matrix to count token co-occurrences.
- Simultaneous consideration of horizontal and vertical token combinations.
- Ability to merge tokens into arbitrary shapes through iterative merging.

Extended Vocabulary: The training process results in an extended vocabulary that merges tokens based on learned patterns. To analyze the impact of vocabulary size on performance, we trained multiple tokenizer versions with extended vocabulary sizes ranging from 1024 to 16384.

4.2 MLLM Training

Base Model: We used the `Llama-3.1-8B` model (Dubey et al., 2024) as our base text LLM.

Token Embedding Expansion: Before training the MLLM, we first expanded the token embedding layers of the base model to accommodate the new image token IDs. For instance, with 8192 VQ-GAN token IDs and a BPE Image Tokenizer vocabulary of size 4096, we expanded the embedding layers from $n \times m$ to $(n + 8192 + 4096) \times m$, where n is the original text embedding dimension and m is the number of embedding layers. We also added extra special tokens to mark the beginning and end positions of images.

Training Process: The training process consisted of two stages:

- *Stage 1: Image Understanding Pretraining (PT).* In this stage, we froze the original text token embeddings and trained only the new image token embeddings using image caption data. This stage used 595K images from CC-3M (Sharma et al., 2018) and 558K from LCS (Liu et al., 2024b).
- *Stage 2: Supervised Fine-Tuning (SFT).* After stage 1, we unfroze all weights and performed full-parameter fine-tuning using complex conversation data with image inputs. This stage used 1.27 million entries from the LLaVA-OneVision Dataset (Li et al., 2024), including General QA (504K), Doc & Chart (249K), Reasoning (343K), and OCR (180K) tasks.

The core distinction between our proposed training pradigm and existing widely used MLLM training approaches (Liu et al., 2024b;a) mainly lies in the first stage of pretraining. The conventional approaches typically align a pretrained CLIP encoder (Radford et al., 2021) with a text-based LLM, where the image understanding capability is largely derived from the CLIP encoder, which was trained on 400 million image-text pairs (Radford et al., 2021), far exceeding the amount of data we provide to our model for image understanding. Though our proposed paradigm also involves pretraining an image tokenizer, it does not directly provide image comprehension capability to the tokenizer. Instead, the MLLM acquires its entire image comprehension capability after the first stage of training. Intuitively, this approach allows for a more direct fusion of the image modality with the text-based LLM, potentially mitigating biases introduced by separate training and subsequent alignment. Our experimental results also can demonstrate that even with a significantly limited amount of data compared to that used in pretraining the CLIP encoder, we can still endow text-based LLMs with good image understanding capabilities.

5 Experimental Results and Analysis

Our experiments aimed to evaluate the effectiveness of the proposed BPE Image Tokenizer and its impact on MLLM performance. We analyze the results from three main perspectives: the impact of the BPE Image Tokenizer, the effect of additional data scaling, and the influence of extended vocabulary. Table 5.1 summarizes our main results, while Figures 3(a) and 3(b) provide insights into the impact of the BPE vocabulary and its token usage patterns.

5.1 Experiments Setup

Benchmarks: We evaluated our model using multiple benchmarks:

- VQAv2 (Goyal et al., 2017): Visual Question Answering
- MMBench (Liu et al., 2023): Multimodal Understanding
- MME (Fu et al., 2023): Multimodal Evaluation (separate tests for perception MME^p and cognition MME^c)
- POPE (Li et al., 2023): Object Hallucination Evaluation
- VizWiz (Gurari et al., 2018): Visual Question Answering for Visually Impaired Users

Data Scaling Experiments: To further investigate the impact of dataset size on performance, we gradually augment our training data with RefCOCO (50.6K) (Kazemzadeh et al., 2014), AOKVQA (66.2K) (Schwenk et al., 2022), ShareGPT4o (57.3K) (Chen et al., 2023), and ALLaVA Inst (70K) (Chen et al., 2024).

Table 5.1: Performance comparison of different settings and training strategies across multiple benchmarks. Consistent improvements across all benchmarks demonstrate the benefit from the BPE Image Tokenizer and the effectiveness of the designed training strategy. Further performance gains are observed with incremental data additions, highlighting the potential scalability of our approach. Specifically, the 'LLM' represents the `Llama-3.1-8B` backbone model we used. MME^p and MME^c represent MME-perception and MME-cognition, respectively.

	Training type	VQAv2	MMBench	MME^p	MME^c	POPE	VizWiz
LLM+VQ	SFT	51.1	35.9	972.3	231.8	73.8	43.1
	PT(full)+SFT	53.7	37.0	1037.2	261.4	75.3	44.2
	PT(freeze)+SFT	55.4	37.6	1054.5	277.0	76.0	45.3
LLM+VQ+BPE	SFT	52.2	35.4	1029.7	269.6	76.3	45.3
	PT(full)+SFT	56.5	38.6	1144.6	284.3	77.3	45.8
	PT(freeze)+SFT	57.1	40.9	1223.5	307.1	79.0	46.0
Additional scaling (PT)	+RefCOCO(50.6K)	58.6	42.3	1257.4	314.3	79.8	47.1
	+AOKVQA (66.2K)	59.6	43.1	1288.1	321.4	80.4	47.5
Additional scaling (SFT)	+ShareGPT4o (57.3K)	60.2	43.7	1304.5	327.7	80.9	47.8
	+ALLaVA Inst (70K)	60.6	44.0	1316.2	331.0	81.3	48.2

5.2 Impact of the BPE Image Tokenizer

Importance of Two-Stage Training: Our results clearly demonstrate the necessity of the two-stage training process. Models trained only with SFT show notably lower performance compared to those that underwent both PT and SFT. This finding underscores the importance of the pretraining stage in guiding the expanded token embedding layers to learn meaningful visual representations.

Freezing *vs*. Full Pretraining: We observe a consistent performance advantage when freezing the text token embeddings during pretraining, *i.e.*, PT(freeze), compared to training all embeddings, *i.e.*, PT(full). For instance, in the 'LLM+VQ+BPE' setting, PT(freeze)+SFT outperforms PT(full)+SFT across all benchmarks, with notable improvements in MME^p (1223.5 *vs*. 1144.6) and MME^c (307.1 *vs*. 284.3). This suggests that allowing the text embeddings to update during pretraining may interfere with the model's ability to focus on learning visual representations.

Performance Gains: The integration of our BPE Image Tokenizer improves model performance across all benchmarks, as evidenced by comparing the 'LLM+VQ' and 'LLM+VQ+BPE' rows in Table 5.1. These gains indicate that the BPE Image Tokenizer provides the model with better multimodal understanding. The improvements is consistent across different training methods, also highlighting the robustness of our approach.

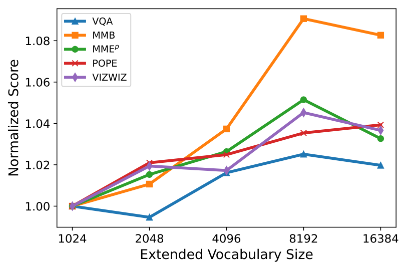
5.3 Impact of Additional Data Scaling

To investigate the scalability of our approach, we incrementally add more datasets to both the pretraining and SFT phases. The results, shown in the lower part of Table 5.1, reveal a clear trend of performance improvement with increased data volume.

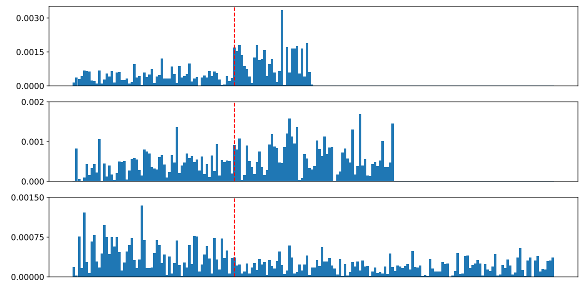
Pretraining Data Scaling: Adding RefCOCO (50.6K) and AOKVQA (66.2K) to the pretraining phase leads to consistent improvements across all benchmarks. For instance, VQAv2 scores increase from 57.1 to 59.6, and MMBench scores rise from 40.9 to 43.1. This suggests that our model benefits from exposure to a wider range of visual concepts and question-answering patterns during pretraining.

SFT Data Scaling: Further improvements are observed when adding ShareGPT4o (57.3K) and ALLaVA Inst (70K) to the SFT phase. The consistent improvement across different benchmarks indicates that our approach effectively leverages additional data to enhance multimodal understanding.

Scalability Potential: The continuous performance improvements with data scaling suggest that our training paradigm has not yet reached its upper limit. This finding is encouraging, as it implies that further performance gains could be achieved with larger datasets or more diverse data sources.



(a) The performance trend on various datasets as the BPE vocabulary size changes. To intuitively compare the trends across different datasets, we normalize the scores on each dataset to the score obtained when using a vocabulary size of 1K. The results show that for most datasets, performance reaches its optimum when the vocabulary size is 8K.



(b) Visualization of token embedding weights with different vocabulary sizes. From top to bottom, they respectively represent the 4K/8K/16K versions. The vertical axis represents the mean absolute value of weights across all layers. The horizontal axis represents the range of corresponding image token IDs. The left side of the red line represents the range of VQ-GAN's codebook (8K). The right area represents the range of correspondingly expanded BPE vocabulary (4K/8K/16K).

Figure 3: (Left) The relationship between model performance and the size of the BPE vocabulary. (Right) The visualization of model weights for tokens usage under different vocabularies.

5.4 Impact of BPE Vocabulary and Token Usage Patterns

We conduct a detailed analysis of how the vocabulary of the BPE Image Tokenizer affects model performance and token usage patterns. Figure 3(a) illustrates the relationship between vocabulary size and normalized performance scores across different benchmarks. We observed that when the vocabulary size is lower than 8K (which equals the codebook size of the VQ-GAN model), the performance improves with the increase in vocabulary size. However, when the vocabulary size reaches 16K, all models on datasets except POPE show performance decline. This trend suggests an optimal vocabulary size should balances efficiency and model complexity.

To further verify this finding and gain deeper insights into how models utilize different vocabularies, we visualize the token embedding weights of models trained with different vocabulary sizes. Figure 3(b) shows the absolute values of these weights, averaged across all layers. We observe three distinct patterns: 1) With a smaller extended vocabulary (4K), the model tends to utilize more extended tokens from the BPE image tokenizer; 2) While with a larger extended vocabulary (16K), the model uses more of the original token IDs covered by the VQ-GAN's codebook; 3) Only when using a balanced extended vocabulary size (8k) does the model's token selection become more uniform, achieving the best performance.

We hypothesize that this phenomenon occurs because the original token IDs covered by the VQ-GAN codebook contain fine-grained visual information, while the combined token IDs provided by the BPE Image Tokenizer capture higher-level semantic concepts. A balanced utilization of these two types of information appears to be most conducive to the model's overall learning and performance. These findings may provide valuable insights for future MLLM training design.

6 Related Work

Recent advancements in Multimodal Large Language Models (MLLMs) have demonstrated remarkable capabilities in various tasks, including visual question answering, image captioning, and cross-modal retrieval (Yin et al., 2023). The evolution of MLLMs can be broadly categorized into two main approaches: late-fusion models with specialized encoders and early-fusion token-based models.

Traditional late-fusion MLLMs typically employ distinct specialized designs for different modalities (Team, 2024). In this approach, modality-specific modules are trained separately using specialized data, followed by additional alignment steps to achieve late-fusion of different modalities. For instance, in the image modality, CLIP-based visual encoders (Radford et al., 2021; Fang et al., 2023) are first pre-trained on extensive image-caption datasets, then aligned with text-only LLM backbones (Touvron et al., 2023; Peng et al., 2023) using additional data. Notable examples of this approach include Flamingo (Alayrac et al., 2022), LLaVA (Liu et al., 2024b;a), IDEFICS (Laurençon et al., 2024), and Emu (Sun et al., 2023). However, these models often face challenges in modality alignment, leading to issues such as hallucination in MLLMs (Bai et al., 2024).

To address these challenges, recent research has explored early-fusion approaches through unified representations, proposing token-based methods for multimodal learning (Team, 2024; Lu et al., 2024; Zheng et al., 2024). These methods typically utilize Vector Quantization (VQ) models (Van Den Oord et al., 2017; Razavi et al., 2019; Esser et al., 2021) to convert images into discrete tokens. The concept of token-based multimodal learning was initially explored in studies such as BEiT (Bao et al., 2021), which introduced a self-supervised method for acquiring visual representations based on tokenized image patches. This idea was further developed in works like Cm3 (Aghajanyan et al., 2022) and CM3Leon (Yu et al., 2023), which enabled joint reasoning across modalities and scaled up to autoregressive text-to-image generation. More recent models like Gemini (Team et al., 2023), Unicode (Zheng et al., 2024), and Chameleon (Team, 2024) have adopted end-to-end token-based approaches for multimodal learning.

While these token-based MLLMs demonstrate enhanced reasoning and generation capabilities across various modalities without requiring modality-specific components, they still face challenges in representation learning and alignment (Baltrušaitis et al., 2018). Our proposed BPE Image Tokenizer paradigm addresses these challenges by adhering more closely to the learning method of text-only LLMs. Unlike current token-based approaches, our method directly incorporates crucial structural prior information into the tokens through explicit merging. This approach enables Transformer models to learn input data more effectively, as supported by both our theoretical analysis and experimental results.

The key distinction of our work lies in its focus on optimizing the tokenization process itself, rather than relying solely on pre-trained visual encoders or simple quantization. While traditional visual encoders can achieve high compression rates, their encoded embeddings often depend on specialized decoders for interpretation. In contrast, our BPE Image Tokenizer creates tokens that are directly meaningful to the Transformer model, facilitating more effective learning of visual information without the need for specialized decoders. This approach bridges the gap between visual and textual modalities more seamlessly, potentially leading to more robust and versatile MLLMs.

7 Conclusions, Limitations and Future Work

In this paper, we proposed a novel paradigm for MLLM training, demonstrating improvements in image understanding capabilities across various benchmarks, even with limited training data. Our approach centers on the novel BPE Image Tokenizer, which effectively combines image token IDs to facilitate better integration of visual information into MLLMs. We provided theoretical insights into the importance of tokenization for learning 2D sequence data with Transformer models, supporting our empirical findings and offering a new perspective on processing visual information in language models. Our experiments not only showcased the effectiveness of the BPE Image Tokenizer but also demonstrated the scalability of our approach.

While our results are promising, we acknowledge several limitations of our current work. The experiments were conducted with limited computational resources and training data compared to state-of-the-art MLLMs, potentially understating the full potential of our approach. Additionally, the simplified 2D Markov process used in our theoretical analysis, while instructive, may not fully capture the complexities of real-world image data.

Based on the findings and limitations, we propose several directions for future work. Scaling up the training data and model size would allow us to fully explore the potential of our BPE Image Tokenizer approach in large-scale MLLMs. Investigating the applicability of our method to other visual tasks, including video understanding, could significantly broaden its impact. Exploring more sophisticated tokenization strategies that can better capture the complex dependencies in real-world multimodal data is another promising avenue.

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.

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, et al. Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*, 2022.

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.

Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. *arXiv preprint arXiv:2404.18930*, 2024.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2): 423–443, 2018.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model, 2024.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 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.

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

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*, pp. 12873–12883, 2021.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19358–19369, 2023.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

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

Yang Jin, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, et al. Unified language-vision pretraining with dynamic discrete visual tokenization. *arXiv preprint arXiv:2309.04669*, 2023.

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.

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne 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*, 2023. URL https://openreview.net/forum?id=xozJw0kZXF.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.

Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26439–26455, 2024.

Ashok Vardhan Makkuva, Marco Bondaschi, Adway Girish, Alliot Nagle, Martin Jaggi, Hyeji Kim, and Michael Gastpar. Attention with markov: A framework for principled analysis of transformers via markov chains. *arXiv preprint arXiv:2402.04161*, 2024.

Mike A Merrill, Mingtian Tan, Vinayak Gupta, Tom Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. *arXiv preprint arXiv:2404.11757*, 2024.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

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.

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.

Nived Rajaraman, Jiantao Jiao, and Kannan Ramchandran. Toward a theory of tokenization in llms. *arXiv preprint arXiv:2404.08335*, 2024.

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.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pp. 146–162. Springer, 2022.

Rico Sennrich. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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.

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

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2(3), 2023.

Qiming Zhang, Jing Zhang, Yufei Xu, and Dacheng Tao. Vision transformer with quadrangle attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Sipeng Zheng, Bohan Zhou, Yicheng Feng, Ye Wang, and Zongqing Lu. Unicode: Learning a unified codebook for multimodal large language models. *arXiv preprint arXiv:2403.09072*, 2024.

A Theoretical Discussions

A.1 Full Proofs

Proof of Proposition 1. We begin by proving equation 3. For any unigram model $Q \in \mathcal{Q}_{\text{1-gram}}$ on the tokens, there exist distributions $Q_{\#}$ and Q_{tok}, supported on $\mathbb{N}$ and Dict (dictionary of the tokenizer) respectively, such that for all token sequence $\boldsymbol{t} = (t_1, ..., t_{|\boldsymbol{t}|})$,

$$Q(\boldsymbol{t}) = Q_{\#}(|\boldsymbol{t}|) \prod_{r=1}^{|\boldsymbol{t}|} Q_{\text{tok}}(t_r).$$

Consequently, for every $m \in \mathbb{N}$ we have

$$\begin{aligned}
\mathcal{L}_m(Q) &= -\frac{1}{m^2}\mathbb{E}\left[\log Q_{\#}(|\operatorname{flat}(\boldsymbol{X})|)\right] - \frac{1}{m^2}\mathbb{E}\left[\sum_{t \in \operatorname{flat}(\boldsymbol{X})} \log Q_{\text{tok}}(t)\right] \\
&\overset{(i)}{=} -\frac{1}{m^2}\mathbb{E}\left[\log Q_{\#}(m^2)\right] - \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbb{E}\left[\log Q_{\text{tok}}(X_{ij})\right] \\
&\geq -\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathbb{E}\left[\log Q_{\text{tok}}(X_{ij})\right] \\
&\overset{(ii)}{=} -\sum_{a \in \mathcal{C}} \pi(a)\log(Q_{\text{tok}}(a)) \\
&\geq H(\pi),
\end{aligned}$$

where in (i) we apply the definition of $\operatorname{flat}(\cdot)$, and in (ii) we use the fact that π is the stationary distribution for each column/row as assumed in Scenario 1/Scenario 2. Thus, equation 3 is proved.

Next, we proceed to prove equation 4. For an arbitrary model Q, we have

$$\begin{aligned}
m^2\mathcal{L}_m(Q) &= -\mathbb{E}[\log(Q(\operatorname{flat}(\boldsymbol{X})))] \\
&= \mathbb{E}[\log(P^*(\operatorname{flat}(\boldsymbol{X}))/Q(\operatorname{flat}(\boldsymbol{X})))] - \mathbb{E}[\log(P_m^*(\operatorname{flat}(\boldsymbol{X}))] \\
&= D_{\text{KL}}(P_m^* || Q) + H(P_m^*), \\
&\geq H(P_m^*).
\end{aligned}$$

where P_m^* denotes the actual joint distribution of $\operatorname{flat}(\boldsymbol{X})$ and $D_{\text{KL}}(P_m^*||Q)$ denotes the KL divergence between P_m^* and Q. Under Scenario 1, leveraging the independence between columns we derive

$$\begin{aligned}
&H(P_m^*) \\
&= -\sum_{j=1}^{m}\left\{\mathbb{E}[\log \pi(X_{1j})] + \sum_{i=2}^{m}\mathbb{E}[\log P(X_{ij} \mid X_{i-1,j})]\right\} \\
&= -\sum_{j=1}^{m}\left\{\sum_{a \in \mathcal{C}}\pi(a)\log(\pi(a)) + (m-1)\sum_{a \in \mathcal{C}}\sum_{a' \in \mathcal{C}}\pi(a)P(a' \mid a)\log\left(P(a' \mid a)\right)\right\} \\
&= mH(\pi) + m(m-1)H_{\infty}.
\end{aligned}$$

This result is also applicable under Scenario 2 by switching the roles of i and j. As a result, in both scenarios we have

$$\liminf_{m \to \infty} \min_{Q} \mathcal{L}_m(Q) \geq \lim_{m \to \infty} \frac{1}{m^2} H(P_m^*) = H_{\infty}.$$

Conversely, it is evident that

$$\min_{Q} \mathcal{L}_m(Q) \leq \mathcal{L}_m(P_m^*) = \frac{1}{m^2}H(P_m^*).$$

Hence,

$$\limsup_{m \to \infty} \min_{Q} \mathcal{L}_m(Q) \leq \lim_{m \to \infty} \mathcal{L}_m(P_m^*) = H_{\infty}.$$

This finishes the proof of equation 4. $\square$

Proof of Proposition 2. In the one-dimensional case, where the data is a string generated from an ergodic Markov process, authors in (Rajaraman et al., 2024) have proved the following result:

Lemma A.1 (Theorem 4.1 in (Rajaraman et al., 2024))**.** *Suppose a string $\boldsymbol{s}$ of length m is generated from an ergodic data source supported on a finite set $\mathcal{A}$, with a transition kernel $P(\cdot \mid \cdot)$ and a stationary distribution π. Assume the transition kernel satisfies $\min_{a,a' \in \mathcal{A}} P(a' \mid a) \triangleq \delta > 0$. Then, there exists a tokenizer with a dictionary containing at most D tokens, along with an encoding function $\mathrm{enc}(\cdot)$ applied to $\boldsymbol{s}$, such that*

$$\begin{aligned} &\limsup_{m\to\infty} \min_{Q \in \mathcal{Q}_{\text{1-gram}}} \frac{1}{m}\mathbb{E}[\log(1/Q(\mathrm{enc}(\boldsymbol{s})))] \\ &\qquad \leq \frac{1}{1-\varepsilon} \sum_{a\in\mathcal{A}} \sum_{a'\in\mathcal{A}} \pi(a) P(a' \mid a) \log\left(1/P(a' \mid a)\right), \end{aligned} \tag{6}$$

where $\varepsilon = \log(1/\delta)/(0.99\log(D))$ and $D \in \mathbb{N}$ is an arbitrary constant that is sufficiently large.

Building on the results of Lemma A.1, we extend these findings to images generated under Scenario 1 or 2.

In both scenarios, the generated image is composed of m parallel Markov sequences, each of length m, sharing the same transition kernel $P(\cdot \mid \cdot)$ and stationary distribution π.

Consider Scenario 1. According to Lemma A.1 and equation 6, there exists a tokenizer equipped with a dictionary, denoted as Dict where $|\,\mathrm{Dict}\,| \leq D$, and an encoding function, denoted as $\mathrm{enc}_{\mathrm{col}}(\cdot)$, and there also exists a unigram model $Q_m \in \mathcal{Q}_{\text{1-gram}}$ for each $m \in \mathbb{N}$, such that

$$\begin{aligned} &\limsup_{m\to\infty} \frac{1}{m}\mathbb{E}[\log(1/Q_m(\mathrm{enc}_{\mathrm{col}}(\boldsymbol{X}_{:j})))] \\ &\leq \frac{1}{1-\varepsilon} \sum_{a\in\mathcal{C}} \sum_{a'\in\mathcal{C}} \pi(a) P(a' \mid a) \log\left(1/P(a' \mid a)\right) \\ &= \frac{1}{1-\varepsilon} H_\infty. \end{aligned}$$

Here $\boldsymbol{X}_{:j}$ represents the j-th column of image $\boldsymbol{X}$ and $j \in \{1, ..., m\}$ is arbitrary. For the unigram model Q_m, there exist distributions $Q_m^{\#}$ and Q_m^{tok}, supported on $\mathbb{N}$ and Dict respectively, such that for all token sequence $\boldsymbol{t} = (t_1, ..., t_{|\boldsymbol{t}|})$,

$$Q_m(\boldsymbol{t}) = Q_m^{\#}(|\boldsymbol{t}|) \prod_{r=1}^{|\boldsymbol{t}|} Q_m^{\mathrm{tok}}(t_r).$$

Consequently, it is straightforward to derive:

$$\limsup_{m\to\infty} \frac{1}{m}\mathbb{E}\left[\sum_{t \in \mathrm{enc}_{\mathrm{col}}(\boldsymbol{X}_{:j})} \log(1/Q_m^{\mathrm{tok}}(t))\right] \leq \frac{1}{1-\varepsilon} H_\infty. \tag{7}$$

To construct a qualified tokenizer for image $\boldsymbol{X}$, we can simply use the dictionary Dict, and define the encoding function as

$$\mathrm{enc}(\boldsymbol{X}) = (\mathrm{enc}_{\mathrm{col}}(\boldsymbol{X}_{:1}), ..., \mathrm{enc}_{\mathrm{col}}(\boldsymbol{X}_{:m})),$$

which concatenates the tokens returned by each column. Subsequently, let $\tilde{Q}_m \in \mathcal{Q}_{\text{1-gram}}$ be the unigram model defined by

$$\tilde{Q}_m(\boldsymbol{t}) = Q_m^{\mathrm{unif}}(|\boldsymbol{t}|) \prod_{r=1}^{|\boldsymbol{t}|} Q_m^{\mathrm{tok}}(t_r).$$

Here Q_m^{unif} is the uniform distribution over $\{1, 2, ..., m^2\}$, and Q_m^{tok} is as previously defined. It then follows that:

$$\begin{aligned}
&\mathcal{L}_m(\tilde{Q}_m \circ \text{enc}) \\
&= \frac{1}{m^2}\mathbb{E}[\log(1/Q_m^{\text{unif}}(|\,\text{enc}(\boldsymbol{X})|)) + \frac{1}{m^2}\mathbb{E}\left[\sum_{j=1}^{m}\sum_{t\in\text{enc}_{\text{col}}(\boldsymbol{X}_{:j})}\log(1/Q_m^{\text{tok}}(t))\right] \\
&= \frac{2\log(m)}{m^2} + \frac{1}{m}\mathbb{E}\left[\sum_{t\in\text{enc}_{\text{col}}(\boldsymbol{X}_{:j})}\log(1/Q_m^{\text{tok}}(t))\right]
\end{aligned}$$

Finally, by equation 7

$$\limsup_{m\to\infty}\min_{Q\in\mathcal{Q}_{\text{1-gram}}}\mathcal{L}_m(Q\circ\text{enc}) \le \limsup_{m\to\infty}\mathcal{L}_m(\tilde{Q}_m\circ\text{enc}) \le \frac{1}{1-\varepsilon}H_\infty,$$

thereby proving equation 5.

The proof of Scenario 2 is quite similar and is omitted here. □

A.2 Information Loss of the Tokenizer

Let $\boldsymbol{X}$ denote the original image, $\boldsymbol{V}$ denote the VQ-tokenized sequence, and $\boldsymbol{T}$ denote the final BPE-tokenized sequence. The information loss introduced by the BPE tokenization process can be quantified using conditional entropy:

$$L_{bpe} = H(\boldsymbol{X}|\boldsymbol{T}) - H(\boldsymbol{X}|\boldsymbol{V}). \tag{8}$$

This quantity represents the increase in uncertainty about the original image $\boldsymbol{X}$ when using BPE tokens $\boldsymbol{T}$ compared to using VQ tokens $\boldsymbol{V}$. Since

$$L_{bpe} = H(\boldsymbol{X}|\boldsymbol{T}) - H(\boldsymbol{X}|\boldsymbol{V}) = [H(\boldsymbol{X},\boldsymbol{T}) - H(\boldsymbol{T})] - [H(\boldsymbol{X},\boldsymbol{V}) - H(\boldsymbol{V})], \tag{9}$$

and recall that mutual information can be expressed as:

$$\begin{aligned}
I(\boldsymbol{X};\boldsymbol{V}) = H(\boldsymbol{X}) + H(\boldsymbol{V}) - H(\boldsymbol{X},\boldsymbol{V}), \\
I(\boldsymbol{X};\boldsymbol{T}) = H(\boldsymbol{X}) + H(\boldsymbol{T}) - H(\boldsymbol{X},\boldsymbol{T}).
\end{aligned} \tag{10}$$

Substituting back into equation 9 and we get

$$\begin{aligned}
L_{bpe} &= [H(\boldsymbol{X}) + H(\boldsymbol{T}) - I(\boldsymbol{X};\boldsymbol{T}) - H(\boldsymbol{T})] - [H(\boldsymbol{X}) + H(\boldsymbol{V}) - I(\boldsymbol{X};\boldsymbol{V}) - H(\boldsymbol{V})] \\
&= [H(\boldsymbol{X}) - I(\boldsymbol{X};\boldsymbol{T})] - [H(\boldsymbol{X}) - I(\boldsymbol{X};\boldsymbol{V})] \\
&= I(\boldsymbol{X};\boldsymbol{V}) - I(\boldsymbol{X};\boldsymbol{T}).
\end{aligned} \tag{11}$$

Therefore, the information loss can be expressed as the reduction in mutual information between the original image and its tokenized representation. This formulation is particularly useful as it directly quantifies how much information about the original image is preserved through the tokenization process.

Next, we need to analyze how each BPE merge operation affects the mutual information $I(\boldsymbol{X};\boldsymbol{T})$. For any given BPE token sequence $\boldsymbol{T}$, we can decompose $H(\boldsymbol{X}|\boldsymbol{T})$ based on individual tokens:

$$H(\boldsymbol{X}|\boldsymbol{T}) = \sum_t P(t)H(\boldsymbol{X}|\boldsymbol{T}=t). \tag{12}$$

During a merge operation that combines tokens (t_i, t_j) into t_m, the change in conditional entropy can be expressed as:

$$\Delta H = P(t_m)H(\boldsymbol{X}|\boldsymbol{T}=t_m) - [P(t_i,t_j)H(\boldsymbol{X}|\boldsymbol{T}=t_i,t_j)]. \tag{13}$$

For the whole token distribution, the increase in conditional entropy can be quantified using the KL divergence:

$$H(\boldsymbol{X}|t_m) = H(\boldsymbol{X}|t_i,t_j) + \mathrm{KL}(P(\boldsymbol{X}|t_i,t_j)||P(\boldsymbol{X}|t_m)), \tag{14}$$

where $\mathrm{KL}(\cdot)$ represents the KL divergence. In this equation,

- $H(\boldsymbol{X}|t_i,t_j)$ represents the original uncertainty about $\boldsymbol{X}$ given the separate tokens
- $\mathrm{KL}(P(\boldsymbol{X}|t_i,t_j)||P(\boldsymbol{X}|t_m))$ represents the extra uncertainty introduced by merging the tokens
- The equation shows that no other sources of information loss exist beyond these two terms.

This leads to an initial upper bound on the total information loss:

$$L_{bpe} \leq \sum_{merges} \mathrm{KL}(P(\boldsymbol{X}|t_i,t_j)||P(\boldsymbol{X}|t_m)). \tag{15}$$

The summation here represents the sum over all merges during the BPE process. Considering that BPE performs merges based on frequency patterns and with minimum merge frequency threshold $p_{\min}$, we can relax the inequality to get a more interpretable bound:

$$L_{bpe} \leq (|D_{vq}| - |D_{bpe}|) \times (-p_{\min}\log(p_{\min})). \tag{16}$$

Here, $|D_{vq}|$ is the size of the VQ codebook, and $|D_{bpe}|$ is the size of the vocabulary after BPE extension.

To put this bound in perspective, consider a typical configuration:

- $|D_{vq}| = 8192$ (VQ codebook size)
- $|D_{bpe}| = 8192 + 8192$ (vocabulary size after BPE extension)
- $p_{\min} = 0.01$ (minimum merge frequency)

The upper bound on information loss for the whole vocabulary would be:

$$L_{bpe} \leq (8192 + 8192 - 8192) \times (-0.01 \times \log(0.01)) \approx 377.3 \text{ bits}. \tag{17}$$

For a single image, the original VQ tokens (32×32 patches) contain $32 \times 32 \times \log_2 8192 = 13312$ bits information, and the per token loss of the extended BPE vocabulary is $L_{bpe}/(|D_{vq}| - |D_{bpe}|) \approx 0.046$ bits. We can calculate that the max loss ratio of the single image is only $32 \times 32 \times 0.046/13312 \approx 0.35\%$, which is a relatively small information loss. Considering the benefits brought by BPE tokenization as discussed earlier, we believe this loss is acceptable.

B Implementation Details

B.1 Additional Pseudo Codes

The additional functions used in Algorithm 4.1 are shown in Algorithm B.1 and B.2.

Algorithm B.1 Update Adjacency Matrix

```
procedure UpdateMatrix(D)
    A ← zeros(v, v)
    for each d ∈ D do
        if d is 1D then
            for j ← 1 to |d| − 1 do
                A[d_j, d_{j+1}] ← A[d_j, d_{j+1}] + 1
            end for
        else
            for each (x, y) in d do
                Update A for neighbors of (x, y)
            end for
        end if
    end for
    return A
end procedure
```

Algorithm B.2 Find Most Frequent Pair

```
procedure MaxFreqPair(A)
    U ← UpperTri(A)
    if ∑U = 0 then
        return (null, 0)
    end if
    (i, j) ← arg max U
    return ((i, j), U_ij)
end procedure
```

B.2 Hyperparameters for Figure 2(b) and 2(c)

In the experiment shown in Figure 2(b), we experimented all parameter combinations listed in Table B.1 for the Transformer model. For the experiment in Figure 2(c) that used a BPE tokenizer, except for the layers shown, all other parameters used the minimum values from Table B.1.

Table B.1: Hyperparameters for Transformer models

Hyperparameter	**Value / Range**
BPE dictionary size	10
batch size	{8, 16, 32}
gradient acc. steps	1
learning rate	2e-3
weight decay	1e-3
scheduler	cosine
optimizer	adamw
dropout	0
number of heads	{1, 2, 4, 8, 16}
number of layers	{2, 4, 6, 8}
embedding dimension	{10, 20, 30, 40}

B.3 Hyperparameters for the VQ-GAN model

The hyperparameters for the VQ-GAN model used in our experiments are shown in Table B.2.

Table B.2: Hyperparameters for the VQ-GAN model

Hyperparameter	Value
embedding dimension	256
codebook size	8192
z_channels	256
resolution	512
dropout	0

B.4 Hyperparameters for training MLLM

The hyperparameters for pre-training / supervised-fine-tuning MLLM are shown in Table B.3.

Table B.3: Hyperparameters for training MLLM

Hyperparameter	PT	SFT
batch size	1	1
gradient accumulation	2	4
learning rate	1e-3	3e-5
learning schedule	cosine	cosine
warmup ratio	0.03	0.03
weight decay	0	0
epoch	3	2
optimizer	AdamW	AdamW
deepspeed stage	2	2

B.5 Details of the Pipeline

In this section, we provide more detailed descriptions of our pipelines as below.

B.5.1 LLM+VQ+BPE Pipeline

Our complete pipeline (LLM+VQ+BPE) consists of three main components:

- Image Quantization: We employ a pretrained VQ-GAN model (with a codebook size of 8192) to quantize input images into a sequence of token IDs. This process converts images of any size into a 1024-dimensional ID tensor.
- BPE Image Tokenizer: This component, which distinguishes our approach from previous methods, processes the quantized token IDs using our trained BPE tokenizer. The tokenizer learns to merge frequently co-occurring tokens, creating a new vocabulary that captures higher-level visual patterns.
- Multimodal Learning: The processed image token IDs are combined with text token IDs and fed into the LLM (Llama-3.1-8B in our experiments). Before training, we expand the token embedding layers of the base model to accommodate both the VQ-GAN token IDs (8192) and the BPE vocabulary (e.g., 4096), along with special tokens for marking image positions.

B.5.2 LLM+VQ BASELINE

The LLM+VQ variant serves as our baseline, following a simpler approach:

- It uses the same VQ-GAN model for image quantization as in LLM+VQ+BPE.
- The quantized token IDs are directly combined with text token IDs, bypassing the BPE tokenizer stage.
- The token embedding layers are only expanded to accommodate the VQ-GAN token IDs and special tokens.

B.5.3 TRAINING PROCESS

Both variants undergo the same two-stage training process:

- Image Understanding Pretraining (PT): During this stage, we train the expanded token embeddings using image caption data. In the PT(freeze) setting, we freeze the original text token embeddings, while in PT(full), all embeddings are updated.
- Supervised Fine-Tuning (SFT): All parameters are unfrozen and fine-tuned using SFT data.

C EXAMPLES

Table C.1: Output examples of models trained with and without using BPE tokenizer. The text marked in red has obvious hallucinations.

Case	LLM+VQ+BPE	LLM+VQ
Q: Give a brief description of the image. 	The image appears to be taken from the perspective of someone riding a bicycle or motorcycle on a dirt road or mountain trail. We can see handlebars in the foreground, suggesting this is a first-person view from the rider.	The image features a motorcyclist riding on a dirt road, wearing a helmet and a black jacket. They are surrounded by a group of people who are taking pictures of the rider as they pass by. The motorcyclist appears to be enjoying the moment and capturing the memory.
Q: What is in the photo? 	This photo shows a workspace or home office setup. There are two computer monitors on a desk. The desk is positioned in front of a window. The desk appears to be an L-shaped or corner desk, with a clean and minimalist design.	In the photo, a desk is set up with a laptop positioned in the center, surrounded by a few other laptops and monitors. There is an empty black chair placed in front of the desk. A potted plant is located on the left side of the scene, creating a lively atmosphere.

As shown in Table C.1, in these two cases, the version without using the BPE tokenizer, which directly uses VQ-quantized tokens for `Llama-3.1-8B` training, frequently exhibits hallucination issues. In contrast, the model trained with the BPE tokenizer can more accurately describe the information in the images, showcasing better image comprehension capabilities.

D Broader Impact

The development of our proposed noval paradigm for training MLLMs has potential to advance artificial intelligence and its applications across various domains, including medical imaging and autonomous systems. However, this progress also raises important ethical considerations, such as privacy concerns and the potential for bias in AI systems. As this technology evolves, it is crucial to balance innovation with responsible development, necessitating collaboration between researchers, ethicists, and policymakers. Future work should not only focus on technical improvements but also on ensuring that these advancements benefit society while mitigating potential risks. This research thus opens new avenues for AI capabilities while underscoring the importance of ethical considerations in technological progress.

E Computing Resources

We list the hardware resources used in Table E.1.

Table E.1: Computing resources for our experiments.

CPU	GPU	RAM
Intel 3GHz $\times$ 64	Nvidia A800 (80GB) $\times$ 8	1024GB

F Licenses

In our code, we have used the following libraries which are covered by the corresponding licenses:

- Numpy (BSD-3-Clause license)
- PyTorch (BSD-3-Clause license)
- Transformers (Apache license)
- Numba (BSD-2-Clause license)

G Additional Results

G.1 Additional Training with Llama2 version / freezing during SFT

Table G.1: Full results with additional experiments. The red parts indicate using Llama2 as the base LLM for training. The blue parts indicate freezing the embedding corresponding to the visual token during the SFT stage.

	Training type	**VQAv2**	**MMBench**	**MME^p**	**MME^c**	**POPE**	**VizWiz**
Llama3.1+VQ	SFT	51.1	35.9	972.3	231.8	73.8	43.1
	PT(full)+SFT	53.7	37.0	1037.2	261.4	75.3	44.2
	PT(freeze text)+SFT	55.4	37.6	1054.5	277.0	76.0	45.3
Llama2+VQ	PT(freeze text)+SFT	54.0	35.7	991.9	254.2	75.1	44.4
Llama3.1+VQ+BPE	SFT	52.2	35.4	1029.7	269.6	76.3	45.3
	PT(full)+SFT(freeze visual)	31.7	17.8	624.1	171.9	46.4	29.5
	PT(full)+SFT	56.5	38.6	1144.6	284.3	77.3	45.8
	PT(freeze text)+SFT(freeze visual)	22.5	13.3	488.7	143.6	35.2	21.5
	PT(freeze text)+SFT	57.1	40.9	1223.5	307.1	79.0	46.0
Llama2+VQ+BPE	PT(freeze text)+SFT	56.5	38.1	1112.2	277.8	77.9	44.9
Additional scaling (PT)	+RefCOCO(50.6K)	58.6	42.3	1257.4	314.3	79.8	47.1
	+AOKVQA (66.2K)	59.6	43.1	1288.1	321.4	80.4	47.5
Additional scaling (SFT)	+ShareGPT4o (57.3K)	60.2	43.7	1304.5	327.7	80.9	47.8
	+ALLaVA Inst (70K)	60.6	44.0	1316.2	331.0	81.3	48.2

G.2 ADDITIONAL EVALUATION

Table G.2: Comparison of LLM+VQ+BPE and LLM+VQ performance across different categories in MME benchmark.

	Category	**LLM+VQ+BPE**	**LLM+VQ**
Perception	Existence	**145.00**	113.33
	Count	**120.00**	110.00
	Position	**106.67**	103.33
	Color	**148.33**	120.00
	Posters	**136.24**	121.08
	Celebrity	**111.76**	89.51
	Scene	**125.00**	101.75
	Landmark	**130.25**	110.00
	Artwork	**112.75**	90.50
	OCR	87.50	**95.00**
Cognition	Commonsense Reasoning	**107.14**	89.57
	Numerical Calculation	**62.50**	50.00
	Text Translation	95.00	**102.50**
	Code Reasoning	**42.50**	35.00

Table G.3: Comparison of LLM+VQ+BPE and LLM+VQ performance across different categories in MLLM-bench.

Category	**LLM+VQ+BPE**	**LLM+VQ**	**Tie**
perception	26	**34**	10
understanding	**52**	33	25
Applying	**27**	14	19
Analyzing	**49**	40	31
Evaluation	12	**19**	9
Creation	7	**9**	4
Total	**173**	149	98