ObjecTok: Learning Holistic and Robust Object Tokens for MLLMs

Sihan Wang,[1, 2] **Xiyao Liu,**[1*] **Lianqing Liu,**[1] **Zhi Han**[1]

[1]State Key Laboratory of Robotics and Intelligent Systems, Shenyang Institute of Automation, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
{wangsihan, liuxiyao, lqliu, hanzhi}@sia.cn

Abstract

Mainstream multimodal large language models (MLLMs) rely on patch-based tokenization methods, which compromise the integrity of objects and thereby limit the model's perception capabilities while triggering object-related hallucinations. To address this issue, we propose ObjecTok, an innovative object tokenization framework. ObjecTok generates a single, holistic object token for each object in an image. This token is produced by a specially trained object encoder that embeds the object's semantic, positional, and shape information into a single compact representation, thereby preserving the object's integrity. To mitigate the imperfections of upstream object proposer models, we introduce learnable confidence embeddings. These embeddings enable the MLLM to learn the reliability of each object's information, significantly enhancing the model's robustness. Additionally, ObjecTok employs a hybrid input strategy, combining object tokens with traditional image patch tokens, allowing the model to leverage both object-level information and global scene context. By integrating ObjecTok into the LLaVA architecture, we achieve notable performance improvements on multiple object-centric benchmarks, effectively reducing object hallucinations and enhancing perception capabilities. Experimental results robustly demonstrate that the object tokens generated by our ObjecTok framework hold great potential for building more powerful and reliable MLLMs.

Introduction

In recent years, the integration of visual and linguistic modalities has become a central focus in the development of artificial intelligence, particularly in tasks requiring precise understanding of images and their contextual meanings. Currently, significant achievements have been made in the advancement of Multimodal Large Language Models (MLLMs) (Wang et al. 2024; Zhu et al. 2024; Ye et al. 2023; Liu et al. 2023; Dai et al. 2024; Bai et al. 2025), yet most of them rely on a mainstream patch-based tokenization framework. This framework divides images into fixed-size grid-like patches, which are then converted into token sequences for processing (Dosovitskiy et al. 2021), as shown in Fig. 1. Although this method proves effective in many scenarios, its inherent limitations still exist (Qian et al. 2022). A semantically coherent and complete object, such as a person or a car, is often forcibly split and scattered across multiple independent tokens, disrupting the holistic structure of the object. Meanwhile, larger patches could fuse multiple objects into one token, which causes polysemanticity, this could hinder the effective representation learning (Chen et al. 2025).

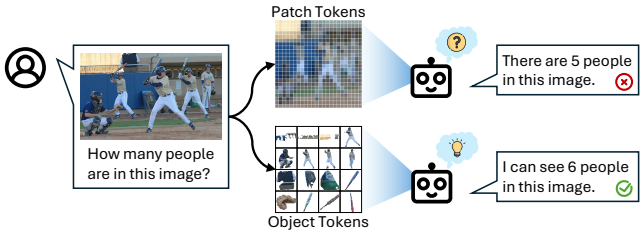


Figure 1: Visualization of different image tokenization methods. Patch tokenization splits images into fixed-sized sequences, which breaks the integrity of the objects. On the other hand, the object tokenization method directly converts all objects within the image into tokens, preserving their semantic and geometric information, so that the MLLM will have a better perception of the objects in the scene.

Patch tokenization not only breaks the morphological structure of the image (Palmer 1977), but also could lead to the loss of critical information, including its high-level semantics, precise spatial position, and distinctive geometric shape. Such informational incompleteness severely limits the model's perception capabilities, resulting in poor performance on tasks like precise object counting (Li et al. 2023b), spatial relationship reasoning (Fu et al. 2023), etc. Moreover, it becomes one of the primary causes of object-related hallucinations (Bai et al. 2024), where the model erroneously generates details that do not exist in the image.

To overcome these limitations, we argue that MLLMs should organize visual inputs into discrete, meaningful object units, rather than arbitrary image patches, much like humans do, as shown in Fig. 1. Based on these inspirations, we propose ObjecTok, an innovative object tokenization framework designed to provide MLLMs with explicit, disentangled representations of individual objects. To achieve this, we utilize an open-vocabulary panoptic segmentation model (Kirillov et al. 2019; Xu et al. 2023) as the upstream object

*Corresponding author

proposer to provide crucial information about objects within images for ObjecTok to convert into object tokens.

We adopt three key features for ObjecTok: holistic object token generation, robustness against segmentation errors and hybrid strategy for preserving global context. *Firstly*, we need to generate a single, holistic object token for each detected object in an image. To achieve this, we designed a novel encoder-decoder scheme to train a dedicated object encoder. This encoder embeds the complete information of each object, including its semantics, position, and shape, into a single compact representation, the object token, thereby preserving the object's integrity. *Secondly*, any method relying on upstream object proposer models must contend with their imperfections. To address this, we introduce learnable confidence embeddings. We transform the confidence scores provided by the object proposer for each object into an embedding vector and incorporate it into the corresponding object token. This allows the MLLM to implicitly learn the reliability of each object's information during training, effectively mitigating the negative impact of imperfect segmentation and significantly enhancing the model's robustness. *Thirdly*, we recognize that relying solely on object information while lacking global context may limit the model's comprehensive reasoning ability. Therefore, ObjecTok adopts a hybrid input strategy, combining our novel object tokens with conventional patch tokens when feeding them into the MLLM. This design enables the model to leverage both information streams synergistically: object tokens provide a structured entity-level view, while patch tokens supplement global scene context and background information, achieving overall complementary advantages.

Finally, to validate the effectiveness of our approach, we integrate the ObjecTok framework into the predominant MLLM architecture, LLaVA (Liu et al. 2023, 2024a,b), to train our own ObjecTok-LLaVA. On multiple object-centric benchmarks (Li et al. 2023b; Fu et al. 2023), our model achieves significant performance improvements. Furthermore, through dedicated robustness tests, we demonstrate that our ObjecTok framework maintains stable high performance even when upstream object proposer's segmentation quality degrades. These results strongly underscore the immense potential of object-level representations (i.e., object tokens) for building more powerful and reliable MLLMs.

In summary, our main contributions are as follows:

- We propose ObjecTok, which shifts visual tokens in MLLMs from fragmented image patches to holistic object-level representations. For each object, we generate a single, compact object token via a specially trained encoder. We designed a dedicated encoder-decoder scheme to train the object encoder, to ensure the object's core semantic, positional, and shape information is preserved.
- To address the inherent imperfections of upstream object proposers, we introduce learnable confidence embeddings. This novel component leverages meta information from the object proposer and allows the MLLM to learn the reliability of each object's information, enhancing the ObjecTok's robustness against upstream errors.
- We combine our proposed object tokens with conventional patch tokens. This hybrid input enables the MLLM to leverage both entity-level details from object tokens and the broader global context provided by patch tokens, resulting in MLLM's stronger perception capabilities.

Related Works

Multimodal Large Language Models

Recent years have witnessed a surge in the development of Multimodal Large Language Models (MLLMs), which extend the impressive capabilities of LLMs to understand and process visual information. Early pioneering works like Flamingo (Alayrac et al. 2022) and the BLIP series (Li et al. 2023a; Dai et al. 2024) established a dominant paradigm. This typically involves using a pre-trained vision encoder (e.g., ViT (Dosovitskiy et al. 2021) or CLIP's vision transformer (Radford et al. 2021)) to extract image features, which are then fed into a large language model via a simple projection layer or a more complex adapter like a Q-Former. Subsequent models, such as LLaVA series (Liu et al. 2023, 2024a), MiniGPT-4 (Zhu et al. 2024), and Qwen-VL series (Bai et al. 2023a), have further advanced this field by leveraging large-scale vision-language instruction tuning datasets, significantly improving their conversational and instruction-following abilities. More recent architectures like CogVLM series (Wang et al. 2024; Hong et al. 2024) have explored deeper fusion mechanisms between the vision and linguistic modalities. While these models have achieved remarkable success, the vast majority still rely on a fixed-size patch representation of the visual input, which is inherently flawed. Our ObjecTok framework aims to address this primary bottleneck of patch-based representations.

Visual Representations in MLLMs

The representation of visual information is a cornerstone of MLLM design. Inspired by the success of Vision Transformers (ViTs) (Dosovitskiy et al. 2021), the patch tokenization method has become the de facto standard. The most predominant visual-language encoders used in MLLMs like CLIP (Radford et al. 2021), EVA-CLIP (Sun et al. 2023) and SigLIP (Zhai et al. 2023), are all derived from ViT. In this approach, an image is divided into a fixed-size grid of non-overlapping patches. Each patch is linearly projected into a vector, which is then treated as a "visual token" by the model. This method is simple and effective for capturing global scene context. However, it is inherently non-object-centric. A single object may be fragmented across multiple patches, while a single patch can contain parts of multiple objects or background clutter (Chen et al. 2025). This spatial and semantic entanglement makes it challenging for MLLMs to perform fine-grained tasks such as precise object counting, localization, or attribute association, often leading to hallucinatory responses (Bai et al. 2024). Some works have attempted to enhance visual representations by increasing image resolution or utilizing a variable number of patches (Bai et al. 2023b; Liu et al. 2024a; Li et al. 2024c; McKinzie et al. 2024). A more recent work, VCoder (Jain, Yang, and Shi 2024), utilizes additional visual inputs, such as ground-truth segmentation and depth maps, to aid

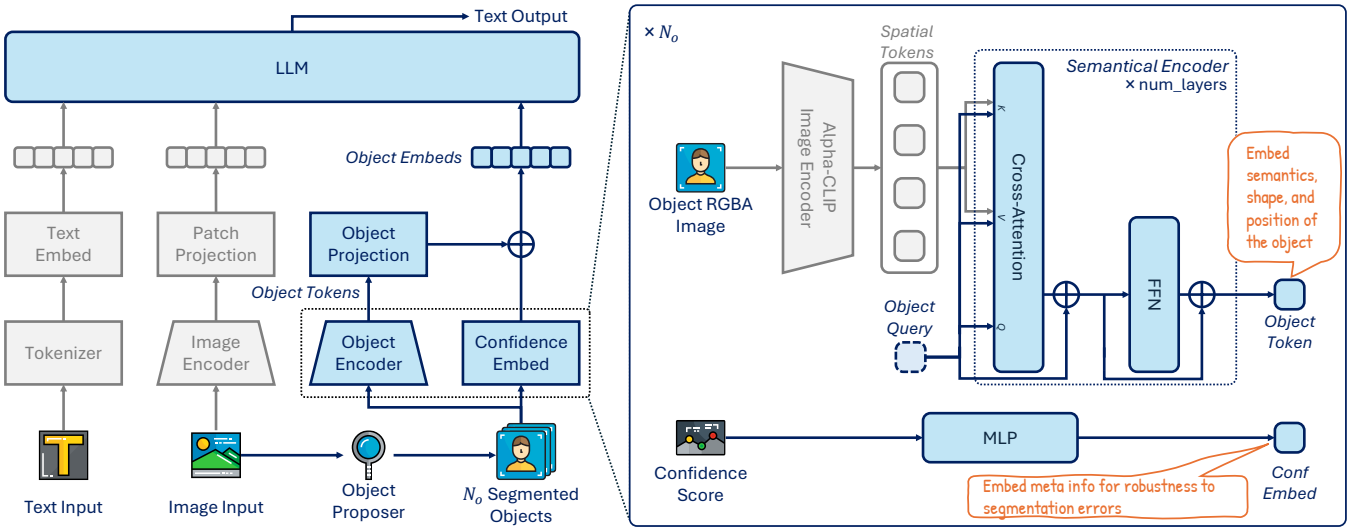


Figure 2: Overall pipeline of our ObjecTok framework. For each object within the input image, we extract an object token that embeds the semantics, shape, and positional information of the object. A confidence embedding, derived from the object's confidence score, is then added for robustness against segmentation errors. In total, an image with N_o objects will result in N_o object embeddings.

MLLMs in perception. Although these methods demonstrate reasonable performance, they do not fundamentally alter the patch-based nature and its associated limitations. Our work, ObjecTok, moves beyond the patch tokens, proposing that representations based on semantically meaningful objects, which is more aligned with object-centric tasks.

Methodology

Overall Architecture

As illustrated in Fig. 2, the ObjecTok framework transforms a single input image into a hybrid sequence of tokens for an MLLM. The overall pipeline consists of four main stages:

1. *Object Proposal*: To extract all objects within the input image, we first employ a pre-trained open-vocabulary panoptic segmentation (Kirillov et al. 2019) model to parse the input image. This model (Xu et al. 2023) outputs a set of masks, each corresponding to a distinct object instance, along with its scalar confidence scores.
2. *Object Tokens Generation*: To convert each detected object into a single, compact object token, we input them as RGBA images into a specially trained object encoder. The object encoder is pre-trained in an encoder-decoder scheme to encapsulate semantic, positional, and shape information of the object into one single object token.
3. *Confidence Embeddings Generation*: To enhance robustness, the confidence scores from the object proposer are converted into learnable confidence embeddings and added to its corresponding object token. This provides the MLLM extra meta information and thus it can develop robustness against upstream segmentation errors.
4. *Integration with MLLMs*: The generated object tokens are combined with the standard patch tokens extracted from the entire image. This hybrid sequence of tokens is passed to the MLLM for response generation. We utilize a dual-stage fine-tuning method for training the MLLM.

We explain each part in detail in the following sections.

Object Proposer

The foundational step of our ObjecTok framework requires a proposer capable of identifying and segmenting objects in a flexible and generalizable manner, without being restricted to a predefined set of categories. To this end, we employ ODISE (Xu et al. 2023), an open-vocabulary panoptic segmentation model, as our object proposer. ODISE is uniquely suited for this task as it leverages the rich internal representations of diffusion models (Rombach et al. 2022) to perform class-agnostic segmentation. This allows it to identify a diverse range of objects and segments within an image, far beyond the scope of closed-set detectors. The object proposer not only generates the masks for each object in the input image, but it also outputs scalar confidence scores; these scores are also an essential part of our ObjectTok framework.

Training Object Encoder

The cornerstone of our framework is the ability to create a single token that holistically represents an object. To achieve this, we design a transformer-based object encoder architecture and train it using an encoder-decoder scheme. The training pipeline of the object encoder is illustrated in Fig. 3.

Object Token Generation For an RGB input image $I^{RGB} \in \mathbb{R}^{H\times W\times 3}$, where H and W are height and width of the image, contains a variable-length set of N_o objects, we have a set of binary masks $M = \{m_i\}_{i=1}^{N_o} \in \{0,1\}^{N_o\times(H\times W)}$ is representing each object. For each ob-

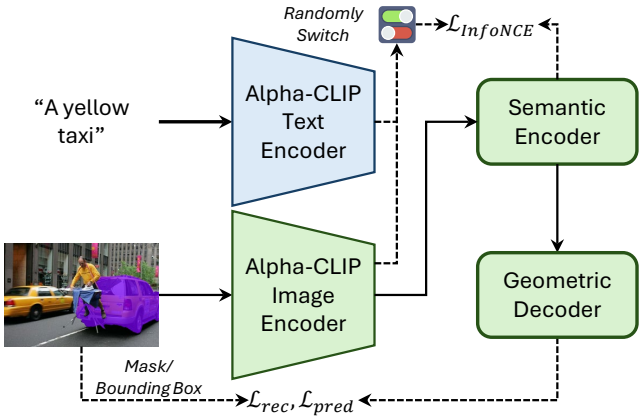


Figure 3: Training pipeline of our object encoder. We learn a holistic representation for each object through an encoder-decoder scheme. The training process has three distinct objectives to ensure that the final object tokens contain semantic, shape, and positional information of the object.

ject O_i, we apply the mask m_i to the alpha channel and create an RGBA image $I_i^{RGBA} \in \mathbb{R}^{H\times W\times 4}$ for the object. We then utilize the pre-trained Alpha-CLIP (Sun et al. 2024) image encoder V_{clip}, which is specifically designed to process RGBA inputs for processing regional semantics, to extract initial features. The encoder outputs a sequence of spatial feature tokens along with a global `[CLS]` token:

$$V_{clip}(I_i^{RGBA}) = \{v_i^{cls}, v_{i,1}^{spatial}, v_{i,2}^{spatial}, \ldots, v_{i,N_a}^{spatial}\}, \tag{1}$$

where v_i^{cls} represents the global semantics of the object and $V_i^{spatial} = \{v_{i,j}^{spatial}\}_{j=1}^{M} \in \mathbb{R}^{N_a\times d_{clip}}$ are the spatial tokens of that object, where N_a is the number of patches and d_{clip} is the embedding size of Alpha-CLIP. The `[CLS]` token contains the overall semantics of the object, and the spatial tokens contain the fine-grained semantics of the object (Qian et al. 2022). However, the spatial tokens are highly sparse (Chen et al. 2024b; Shang et al. 2024; Li et al. 2025; Lin et al. 2025; Yu et al. 2024), which is suboptimal for large models like MLLMs to process. Thus, we need to convert them into a single, holistic object token. To achieve this, we introduce a transformer-based semantic encoder E_{sem}.

The architecture of the semantic encoder, follows the design of Perceiver Resampler (Alayrac et al. 2022). But unlike the original Perceiver Resampler, we zero initialized the learnable object query token $t_i^{obj} \in \mathbb{R}^{1\times d}$ for stable convergence. Overall, the semantic encoder aggregates the spatial tokens into a single object token t_i^{obj}:

$$t_i^{obj} = E_{sem}(V_i^{spatial}, t_i^{obj}). \tag{2}$$

E_{sem} should embed the semantic and geometric information of each input object into corresponding object tokens. To achieve this, we propose a semantic alignment process to retain semantic information and a geometric decoder to encapsulate positional and shape information.

Semantic Alignment To ensure t_i^{obj} is semantically meaningful, we train E_{sem} using a contrastive learning objective. Specifically, we align t_i^{obj} with the high-level semantic representations from Alpha-CLIP. We leverage both the visual `[CLS]` token $v_i^{cls} \in \mathbb{R}^{1\times d_{clip}}$ from the image encoder and the corresponding text `[CLS]` embedding $l_i^{cls} \in \mathbb{R}^{1\times d_{clip}}$ from the text encoder (fed with the object's class name, e.g., "a black cat"). For each t_i^{obj}, we randomly calculate its InfoNCE loss (Oord, Li, and Vinyals 2018) against v_i^{cls} or l_i^{cls}. This ensures that the object tokens not only contain the visual semantic information, but also contain the textual semantic information. The semantic loss is formulated as:

$$\mathcal{L}_{sem}(t_i^{obj}) = \begin{cases} \mathcal{L}_{InfoNCE}(t_i^{obj}, v_i^{cls}) & \text{if } z = 0, \\ \mathcal{L}_{InfoNCE}(t_i^{obj}, l_i^{cls}) & \text{if } z = 1, \end{cases} \tag{3}$$

where $z \sim \text{Bernoulli}(p)$ ($p = 0.5$) indicates the random choice process. In addition, we employ cosine similarity for calculating sample distances in the InfoNCE loss.

Geometric Decoder While $\mathcal{L}_{sem}$ embeds semantic information, it does not explicitly preserve the object's position and shape. To address this, we introduce a dual-stream transformer-based geometric decoder that is trained jointly with the E_{sem}. Its sole purpose is to compel the object token to learn geometric information. To achieve this, the decoder performs two parallel tasks on each stream: masked mask reconstruction for learning shape information and bounding box prediction for learning positional information.

Masked mask reconstruction. Inspired by MAE (He et al. 2022), for each object O_i and corresponding object token t_i^{obj}, we patchify the mask of the object and randomly replace visible tokens $q_i^{vis} \in \mathbb{R}^{N_v\times d}$ with randomly initialized `[MASK]` query tokens $q_i^{mask} \in \mathbb{R}^{N_m\times d}$, where N_v and N_m are the number of visible and masked patches, and $N_t = N_v + N_m$ is the total number of patches. This formulate the reconstruction sequence $Q_i^{rec} = \{q_i^{vis}, q_i^{mask}\} \in \mathbb{R}^{N_t\times D}$. Q_i^{rec} first obtains information from t_i^{obj} through a cross attention mechanism. Then Q_i^{rec} provides information for bounding box prediction through bidirectional attention. Subsequently, q_i^{pred} also provides information for reconstruction through a similar bidirectional attention mechanism. This is because the shape and positional information are relevant, and through this bidirectional information exchange, we ensure that all the necessary information for reconstruction and prediction is fully utilized. Then all `[MASK]` tokens q_i^{mask} are linearly projected to generate reconstructed patches $\hat{A}_i \in \mathbb{R}^{N_m\times(h\times w\times 3)}$, where h and w are the height and width of each patch. We calculate average MSE loss across all masked patches as the objective:

$$\mathcal{L}_{rec} = \frac{1}{N_m}\mathcal{L}_{MSE}(\hat{A}_i, A_i), \tag{4}$$

where $A_i \in \mathbb{R}^{N_m\times(h\times w\times 3)}$ are the ground truth patches. Through this objective, we ensure the object tokens contain the shape information of the corresponding object.

Bounding box prediction. The prediction is performed on another stream of the geometric encoder. To accelerate training convergence, we initialize a detection query token

$q_i^{pred} \in \mathbb{R}^d$ from t_i^{obj}. As described before, q_i^{pred} exchange information with Q_i^{rec} through bidirectional attention mechanism. q_i^{pred} is passed through a two-layer MLP to predict the bounding box $\hat{B}_i \in \mathbb{R}^4$. We then calculate the prediction loss with L1 loss and GIoU loss as the objectives:

$$\mathcal{L}_{pred} = \mathcal{L}_{L1}(\hat{B}_i, B_i) + \mathcal{L}_{GIoU}(\hat{B}_i, B_i), \tag{5}$$

where $B_i \in \mathbb{R}^4$ is the ground truth bounding box corresponding to object O_i. Through this objective, the object token is forced to learn the positional information of the corresponding object, thus fulfilling our intention.

Overall objectives for E_{sem}. The total loss for training the object encoder is a combination of the objectives:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{sem} + \lambda_2 \mathcal{L}_{rec} + \lambda_3 \mathcal{L}_{pred}, \tag{6}$$

where λ_1, λ_2 and λ_3 are the balancing coefficients.

Object Encoder The object encoder contains both Alpha-CLIP image encoder and semantic encoder is defined as our object encoder E_{obj}:

$$t_i^{obj} = E_{obj}(O_i) = E_{sem}(V_{clip}(I_i^{RGBA}), t_i^{obj}). \tag{7}$$

Once the object encoder is trained, it is frozen and integrated into the MLLM pipeline, as shown in Fig. 2. The object token t_i^{obj} is projected through a two-layer MLP to align the embedding size of the base LLM.

Robustness via Confidence Embeddings

The object proposer is imperfect. To make our framework robust to upstream errors, we utilize the confidence score provided by the object proposer. For each object, the scalar confidence score $s_i \in [0, 1]$ is passed through a two-layer MLP to produce a confidence embedding $e_i^{conf} \in \mathbb{R}^{1 \times d_{LLM}}$, where d_{LLM} is the embedding size of the LLM. This embedding is then added to its corresponding object token t_i^{obj} to achieve a final object embedding:

$$e_i^{obj} = t_i^{obj} + e_i^{conf}, \tag{8}$$

where $e_i^{obj} \in \mathbb{R}^{1 \times d_{LLM}}$ is the object embedding. Through the confidence embedding, the MLLM can implicitly learn to down-weight information from low-confidence objects during the training stages. For a image contains N_o objects, we obtain a sequence of object embeddings $S = \{e_i\}_{i=1}^{N_o}$.

ObjecTok-based MLLMs

Integration We recognize that relying solely on object information while lacking global context may limit the model's comprehensive reasoning ability; therefore, we adopt a hybrid tokenization strategy to integrate ObjecTok with MLLMs.Each image is converted into both fixed-size patch tokens and variable-length object tokens, and then projected into the same embedding size with the base LLM for generating responses. Following the setting of LLaVA (Li et al. 2024b; Liu et al. 2024a), the patch tokens are also projected to align with the embedding space of the base LLM.

Thus, we format the input in order of (a) fixed-size patch tokens, (b) variable-length object tokens, and (c) variable-length text tokens. For object tokens, we order them by area. Thus, the prompt template for the final MLLM is "`<img>\n<obj><obj>...<obj>\n<inst>`". The "`<img>`" is replaced with the sequence of patch tokens, and each "`<obj>`" is replaced with the corresponding object token. The "`<inst>`" represents the text instructions.

Training The training of the MLLM with ObjecTok proceeds in two stages inspired by LLaVA (Li et al. 2024b; Liu et al. 2024a): (1) *Pre-training for Alignment*: In the first stage, we use a single-round image captioning dataset. We freeze the visual encoder and our object encoder, as well as the LLM. The primary goal is to adapt the LLM to understand the new vocabulary of object tokens, aligning them with the patch tokens and word embeddings. All projection layers, including confidence embedding layers, are zero-initialized to minimize the disruption caused by the introduction of new modalities. (2) *Instruction Fine-tuning*: In the second stage, we fine-tune the model with a multi-round, multi-modal chat dataset. This stage teaches the MLLM to be more general in visual tasks.

Experiments

Setup

Dataset We use GrIT-20M (Peng et al. 2023) to train the object encoder. GrIT-20m dataset contains ∼20M images with bounding boxes for objects and their text captions. We process the dataset with SAM2 (Ravi et al. 2025). By leveraging the ground truth bounding boxes as prompts, through SAM2, we obtain the masks for the labeled objects.

For training the MLLM with our ObjecTok framework, we build our pre-training and fine-tuning dataset based on ALLaVA-4V (Chen et al. 2024a). ALLaVA-Caption-4V is combined with LLaVA-ReCap-558K dataset (Li et al. 2024a) to be used for pre-training the projection layers and the confidence embedding layers. Then we utilize LLaVA-NeXT-Instruction (Liu et al. 2024b) together with ALLaVA-Instruct-4V for instruction fine-tuning the MLLM. All datasets are processed by ODISE (Xu et al. 2023) to extract objects from images along with their corresponding confidence scores. We treat the pixels that are not labeled in the segmentation results as zero-confidence masks.

Training Details For training the object encoder, we use the balancing coefficients as $\lambda_1 = \lambda_2 = \lambda_3 = 1$, and the embedding size is set to the same as Alpha-CLIP (Sun et al. 2024). Following the set of LLaVA-1.5 (Liu et al. 2024a), we also start from vanilla Vicuna-7B (Chiang et al. 2023) as the base LLM an, and use the same CLIP ViT-L/14-336px (Radford et al. 2021) as the patch-based visual encoder. We call this model **ObjecTok-LLaVA**. We set the learning rate of the confidence embedding layers twice as the learning rate of the projection layers. All models are trained on 8 × NVIDIA A100 80G using DeepSpeed ZeRO-2/3 (Rasley et al. 2020; Rajbhandari et al. 2020), with object encoder trained for 3 epochs and ObjecTok-LLaVA for 1 epoch (both pre-training and fine-tuning stage). We provide more details on hyperparameters in the supplementary materials.

Model	#Params	Random		Popular		Adversarial		Total (Avg. F1)
		Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	
MiniGPT-4	7B	77.83	78.86	68.30	72.21	66.60	71.37	74.15
InsructBLIP	14B	<u>88.73</u>	<u>89.29</u>	81.37	83.45	74.37	78.45	83.73
Qwen-VL	7B	84.37	82.67	84.13	82.06	82.26	80.37	81.70
LLaVA-1.5	7B	88.00	87.14	<u>87.43</u>	<u>86.24</u>	<u>85.50</u>	<u>84.46</u>	<u>85.95</u>
ObjecTok-LLaVA (Ours)	7B	**90.55**	**90.84**	**88.60**	**88.48**	**85.57**	**85.60**	**88.31**

Table 1: Comparison with different MLLMs on POPE benchmark. All results are reported in percentage (%). #Params show the number of each model's parameters. We signify the best and the second best result with **boldface** and <u>underline</u>.

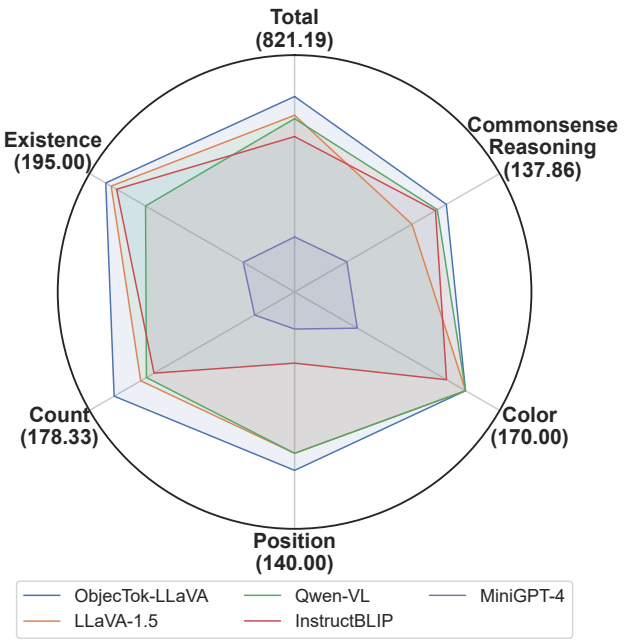


Figure 4: Comparison with different MLLMs on MME benchmark. The chart shows our ObjecTok-LLaVA improves on all evaluated MME tasks, including existence, count, position, color and commonsense reasoning.

Benchmarks We use POPE (Li et al. 2023b) and MME as benchmarks (Fu et al. 2023) to validate our ObjecTok-LLaVA. **POPE** is a benchmark designed to evaluate MLLM's object-centric hallucination level by querying binary-choice questions about objects within images. We report the accuracies and F1 scores on MSCOCO (Lin et al. 2014) tasks. **MME** is a benchmark designed to evaluate MLLM's capabilities across multiple domains comprehensively. Specifically, MME contains object-centric tasks, such as object existence, count, and position, to assess MLLM's object-level hallucination level. We report the scores (based on accuracy) for object existence, count, position, and color tasks, along with an additional evaluation of a commonsense reasoning task to demonstrate the effectiveness of our ObjecTok framework on general perception capabilities.

User	How many uncut fruits are in the image?
LLaVA-1.5	There are four uncut fruits in the image.
ObjecTok-LLaVA	There are three uncut mangosteens in the image.

Table 2: Comparison between outputs of ObjecTok-LLaVA and LLaVA-1.5. In this example, ObjecTok-LLaVA not only outperforms LLaVA-1.5 on counting the number of uncut fruits in the image, but also recognizes the name of the fruit as "mangosteens" accurately.

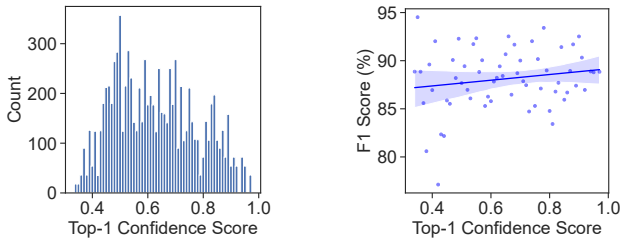
Baselines We compare our ObjecTok-LLaVA with several well-known MLLMs, including MiniGPT-4 (Zhu et al. 2024), InstructBLIP (Dai et al. 2024), Qwen-VL (Bai et al. 2023b) and LLaVA-1.5 (Liu et al. 2024a). These MLLMs have similar scales of parameters and training data sizes to our ObjecTok-LLaVA, thus suitable for comparison.

Results

The results are shown in Tab. 1 and Fig. 4. We give detailed analysis about the results in the following section, and an example of ObjecTok-LLaVA's output in Tab. 2 for reference.

Results on Object Hallucination (POPE) The results are summarized in Tab. 1. Our ObjecTok-LLaVA achieves the highest F1 scores across all three subtasks: Random, Popular, and Adversarial, and culminating in a top overall F1 score of 88.31%. A direct comparison with LLaVA-1.5, which shares the identical 7B base model, signify the contribution of our ObjecTok framework. Our model demonstrates consistent and significant improvements, outperforming LLaVA-1.5 by +3.70%, +2.24%, and +1.14% in F1 scores on the Random, Popular, and Adversarial settings, respectively. This results in a +2.36% uplift in the total average F1 score. These results affirm the effectiveness of our object token in promoting accurate object-centric reasoning.

Results on Comprehensive Perception (MME) As illustrated in Fig. 4, our ObjecTok-LLaVA achieves a leading total score of 821.19, establishing its superior performance



(a) (b)

Figure 5: Robustness test on POPE dataset. (a) The distribution of each sample's top-1 confidence score of all segmented objects within the sample on the POPE dataset. (b) The average F1 score against top-1 confidence score of corresponding image. The results show the performance stability of ObjecTok-LLaVA across varying image qualities.

over strong baselines. A breakdown of the results reveals three key insights. *First*, ObjecTok-LLaVA shows its most pronounced advantages in fundamental perception tasks. It achieves the highest scores in existence, count, and position, which directly validates our hypothesis that the dedicated object encoder effectively captures essential object attributes, including semantic, shape, and positional information. *Second*, and perhaps more importantly, this specialization does not come at the cost of general reasoning. Our model also achieves the top score in commonsense reasoning. This suggests that the structured object information provided by ObjecTok not only improves the perception capabilities of the MLLM on object-centric tasks but also enhances its high-level cognitive abilities, demonstrating the generalizability and broad utility of our approach. *Third*, the improvements on the color tasks are not as significant as in other tasks. We attribute this to a theory that the color tasks are primarily performed by the patch tokens, and we evaluate this hypothesis in the further ablation studies.

Analysis

Robustness Tests In real-world applications, the object proposer is not always perfect. A robust vision-language model should be able to function effectively even with these imperfect, noisy, or incomplete segmentation inputs. To evaluate this crucial capability, we analyze the results of POPE benchmarks. We use the top-1 confidence score provided by the object proposer for object masks within an image as an indicator for the image's overall quality. A low top-1 confidence score generally indicates a higher likelihood of an inaccurate or poor-quality segmentation. We split the samples into bins with a 0.01 top-1 confidence score width. By plotting the F1 score against the top-1 confidence score, we can directly observe how our ObjecTok's performance correlates with the quality of the input segmentation.

The results of our robustness test are presented in Fig. 5, which demonstrates that ObjecTok-LLaVA is highly robust to variations in segmentation quality. Its ability to maintain high performance even with noisy inputs is a critical advantage, making it more practical for real-world scenarios where perfect segmentations cannot be guaranteed.

Model	Existence	Count	Position	Color	Commonsense Reasoning
OT-L	195.00	178.33	140.00	170.00	137.86
w/o GD	170.00	165.00	113.33	151.67	110.71
w/o CE	145.00	156.67	140.00	136.67	107.86
w/o PT	110.00	108.33	93.33	88.33	93.57

Table 3: Results on ablating parts of ObjecTok. *OT-L*: baseline results. *w/o GE*: removing geometric encoder during training object encoder. *w/o CE*: removing confidence embeddings during training MLLM. *w/o PT*: removing patch tokens during training MLLM.

Ablations To dissect the contribution of each key component within our ObjecTok framework, we conduct a series of ablation studies on the MME benchmark. The results are shown in Tab. 3. Overall, we highlight three key insights. *First*, removing the geometric decoder during the object encoder's training phase resulted in a significant performance drop, particularly on position and count tasks. This directly validates our interpretation: the decoder's reconstruction and prediction objectives compel the single object token to co-encode not only semantic information but also crucial spatial information, such as position and shape. *Second*, removing the confidence embeddings resulted in a noticeable decline in overall performance, particularly on object-centric tasks like existence, count and position. This demonstrates the value of this component as a robustness mechanism, as it allows the MLLM to learn to weigh the reliability of each object's information, effectively mitigating the negative impact of imperfect upstream segmentations. *Third*, when we removed the conventional patch tokens and fed only object tokens to the model, performance dropped substantially, especially in commonsense reasoning and color perception. This aligns with our intention of employing a hybrid strategy, where object tokens provide an object-centric view, while patch tokens supply the global context for high-level reasoning. In addition, this also evaluates our hypothesis that the color tasks is mainly carried out by the patch tokens.

Conclusion

In this paper, we introduced ObjecTok, a novel framework designed to address the limitations of patch-based tokenization in MLLMs. By changing the method from patches to object-level representations, ObjecTok generates a single, holistic token for each detected object that embeds its semantic, positional, and shape information. Our proposed hybrid strategy, which combines these object tokens with traditional patch tokens, and learnable confidence embeddings for robustness, allows the MLLM to achieve a more comprehensive and reliable understanding of visual scenes. Extensive experiments demonstrate that ObjecTok significantly enhances performance on object-centric tasks and shows strong resilience to imperfect segmentations. For future work, we aim to extend the ObjecTok concept to the video domain, enabling the MLLMs to track and reason about object interactions better and state changes over time.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62303447 and Grant U23A20343, in part by the Chinese Academy of Sciences through the Project for Young Scientists in Basic Research under Grant YSBR-041, in part by the China Postdoctoral Science Foundation under Grant 2023M743702 and the Postdoctoral Innovation Talents Support Program under Grant BX20230399, and in part by the Natural Science Foundation of Liaoning Province under Grant 2024-MSBA-81, and in part by Fundamental research project of SIA under Grant 2023JC1K01.

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 23716–23736.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023a. Qwen technical report. arXiv:2309.16609.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023b. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. arXiv:2308.12966.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2.5-VL Technical Report. arXiv:2502.13923.

Bai, Z.; Wang, P.; Xiao, T.; He, T.; Han, Z.; Zhang, Z.; and Shou, M. Z. 2024. Hallucination of multimodal large language models: A survey. arXiv:2404.18930.

Chen, D.; Cahyawijaya, S.; Liu, J.; Wang, B.; and Fung, P. 2025. Subobject-level image tokenization. In *ICML*.

Chen, G. H.; Chen, S.; Zhang, R.; Chen, J.; Wu, X.; Zhang, Z.; Chen, Z.; Li, J.; Wan, X.; and Wang, B. 2024a. Allava: Harnessing gpt4v-synthesized data for lite vision-language models. arXiv:2402.11684.

Chen, L.; Zhao, H.; Liu, T.; Bai, S.; Lin, J.; Zhou, C.; and Chang, B. 2024b. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, 19–35.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/. Accessed: 2025-07-03.

Dai, W.; Li, J.; Li, D.; Tiong, A.; Zhao, J.; Wang, W.; Li, B.; Fung, P. N.; and Hoi, S. 2024. InstructBLIP: towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2142–2160.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Fu, C.; Chen, P.; Shen, Y.; Qin, Y.; Zhang, M.; Lin, X.; Yang, J.; Zheng, X.; Li, K.; Sun, X.; et al. 2023. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. arXiv:2306.13394.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *CVPR*, 16000–16009.

Hong, W.; Wang, W.; Ding, M.; Yu, W.; Lv, Q.; Wang, Y.; Cheng, Y.; Huang, S.; Ji, J.; Xue, Z.; et al. 2024. Cogvlm2: Visual language models for image and video understanding. arXiv:2408.16500.

Jain, J.; Yang, J.; and Shi, H. 2024. Vcoder: Versatile vision encoders for multimodal large language models. In *CVPR*, 27992–28002.

Kirillov, A.; He, K.; Girshick, R.; Rother, C.; and Dollár, P. 2019. Panoptic segmentation. In *CVPR*, 9404–9413.

Li, B.; Zhang, H.; Zhang, K.; Guo, D.; Li, F.; Zhang, R.; Liu, Z.; and Li, C. 2024a. LLaVA-NeXT: What Else Influences Visual Instruction Tuning Beyond Data? https://llava-vl.github.io/blog/2024-05-25-llava-next-ablations/. Accessed: 2025-07-03.

Li, B.; Zhang, Y.; Guo, D.; Zhang, R.; Li, F.; Zhang, H.; Zhang, K.; Zhang, P.; Li, Y.; Liu, Z.; et al. 2024b. Llava-onevision: Easy visual task transfer. arXiv:2408.03326.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 19730–19742.

Li, K.; Chen, X.; Gao, C.; Li, Y.; and Chen, X. 2025. Balanced Token Pruning: Accelerating Vision Language Models Beyond Local Optimization. arXiv:2505.22038.

Li, Y.; Du, Y.; Zhou, K.; Wang, J.; Zhao, W. X.; and Wen, J.-R. 2023b. Evaluating object hallucination in large vision-language models. In *EMNLP*, 292–305.

Li, Z.; Yang, B.; Liu, Q.; Ma, Z.; Zhang, S.; Yang, J.; Sun, Y.; Liu, Y.; and Bai, X. 2024c. Monkey: Image resolution and text label are important things for large multi-modal models. In *CVPR*, 26763–26773.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755.

Lin, Z.; Lin, M.; Lin, L.; and Ji, R. 2025. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *AAAI*, 5334–5342.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved baselines with visual instruction tuning. In *CVPR*, 26296–26306.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024b. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/. Accessed: 2025-07-03.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. In *NeurIPS*, 34892–34916.

McKinzie, B.; Gan, Z.; Fauconnier, J.-P.; Dodge, S.; Zhang, B.; Dufter, P.; Shah, D.; Du, X.; Peng, F.; Belyi, A.; et al. 2024. Mm1: Methods, analysis & insights from multimodal llm pre-training. In *ECCV*, 304–323.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748.

Palmer, S. E. 1977. Hierarchical structure in perceptual representation. *Cognitive psychology*, 9(4): 441–474.

Peng, Z.; Wang, W.; Dong, L.; Hao, Y.; Huang, S.; Ma, S.; and Wei, F. 2023. Kosmos-2: Grounding multimodal large language models to the world. arXiv:2306.14824.

Qian, S.; Zhu, Y.; Li, W.; Li, M.; and Jia, J. 2022. What makes for good tokenizers in vision transformer? *IEEE TPAMI*, 11: 13011–13023.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*, 8748–8763.

Rajbhandari, S.; Rasley, J.; Ruwase, O.; and He, Y. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC*, 1–16.

Rasley, J.; Rajbhandari, S.; Ruwase, O.; and He, Y. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD*, 3505–3506.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2025. SAM 2: Segment Anything in Images and Videos. In *ICLR*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*, 10684–10695.

Shang, Y.; Cai, M.; Xu, B.; Lee, Y. J.; and Yan, Y. 2024. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. arXiv:2403.15388.

Sun, Q.; Fang, Y.; Wu, L.; Wang, X.; and Cao, Y. 2023. Eva-clip: Improved training techniques for clip at scale. arXiv:2303.15389.

Sun, Z.; Fang, Y.; Wu, T.; Zhang, P.; Zang, Y.; Kong, S.; Xiong, Y.; Lin, D.; and Wang, J. 2024. Alpha-clip: A clip model focusing on wherever you want. In *CVPR*, 13019–13029.

Wang, W.; Lv, Q.; Yu, W.; Hong, W.; Qi, J.; Wang, Y.; Ji, J.; Yang, Z.; Zhao, L.; XiXuan, S.; et al. 2024. Cogvlm: Visual expert for pretrained language models. In *NeurIPS*, 121475–121499.

Xu, J.; Liu, S.; Vahdat, A.; Byeon, W.; Wang, X.; and De Mello, S. 2023. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2955–2966.

Ye, Q.; Xu, H.; Xu, G.; Ye, J.; Yan, M.; Zhou, Y.; Wang, J.; Hu, A.; Shi, P.; Shi, Y.; et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. arXiv:2304.14178.

Yu, Q.; Weber, M.; Deng, X.; Shen, X.; Cremers, D.; and Chen, L.-C. 2024. An image is worth 32 tokens for reconstruction and generation. In *NeurIPS*, 128940–128966.

Zhai, X.; Mustafa, B.; Kolesnikov, A.; and Beyer, L. 2023. Sigmoid loss for language image pre-training. In *ICCV*, 11975–11986.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2024. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*.