DiViCo: Disentangled Visual Token Compression for Efficient Large Vision-Language Model

Xin Wang, *Member, IEEE*, Zirui Pan, Hong Chen, and Wenwu Zhu, *Fellow, IEEE*

***Abstract*—Large Vision-Language Models have drawn much attention and become increasingly applicable in complicated multimodal tasks such as visual question answering, video grounding, etc. However, it still suffers from inefficiency problem during the inference stage due to the computational overhead brought by the large number of visual tokens. Existing works either utilize an attention score (or visual-text relevance) to filter out the less significant visual tokens, or insert learnable projection layers to directly compress the tokens, which neglects the informative details in visual signals and introduces information loss, resulting in poor generalizability to test data. To solve these problems, in this paper we propose a novel Disentangled Visual Token Compression module, i.e., DiViCo, that effectively compresses the visual tokens and maintains good performance simultaneously. In concrete, we first select the top $\tau\%$ visual tokens according to their average attention scores, then predict the gap between these selected tokens and the original information by employing the chosen tokens in a disentangled and variational manner. Specifically, we model the mean and variance, sampling the predicted gap from the Gaussian prior. We further keep the informativeness of the compressed visual tokens via KL divergence, which ensures the generalizability of the model. Extensive experiments demonstrate the advantage of our proposed DiViCo module against several state-of-the-art baselines over various real-world datasets. Most notably, LLaVA-v1.5-7b equipped with DiViCo is able to reduce 67.7% FLOPs and save 51.7% time while maintaining 95.6% of the accuracy for LLaVA-v1.5-7b without any compression.**

***Index Terms*—Multimodal Representation, Large Vision-Language Model, Token Compression**

I. Introduction

Building on the Large Language Model (LLM) [1], [2], [3], [4], [5], Large Vision-Language Model (LVLM) [6], [7], [8], [9], [10], [11], [12] has achieved revolutionary progress via aligning visual and text modalities to leverage the powerful textual understanding abilities of LLMs. Existing works mainly employ sequential visual representations [12], [10], where visual signals such as images or videos are first divided into patches and then encoded in a series of tokens, which will be projected to the text domain. By resorting to techniques such as visual instruction tuning [11], LVLMs are able to finish complicated multimodal tasks, including image captioning, visual question answering and video grounding, etc.

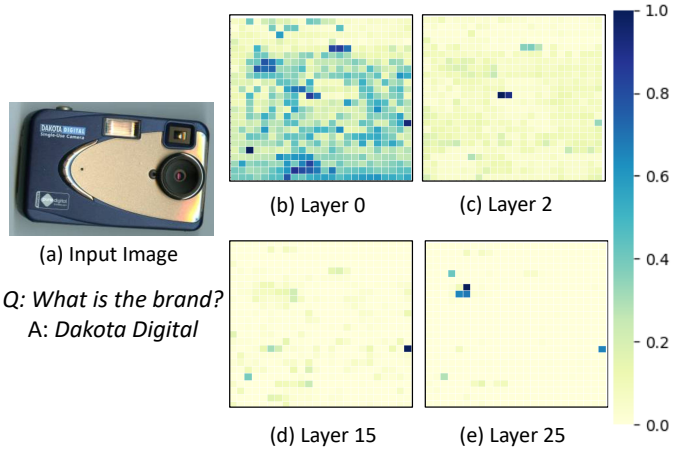


Fig. 1. Average attention scores of the visual tokens at layer K of the decoder, where $K \in \{0, 2, 15, 25\}$. Deeper colors indicate higher scores.

However, the number of visual tokens far exceeds that of text tokens, especially for high-resolution images and videos, resulting in high computational overhead due to the quadratic complexity of the attention mechanism [13]. Moreover, visual tokens also receive lower attention scores than their textual counterparts, contributing less than the textual tokens in LVLMs [14]. As such, we discover that existing LVLMs handle the visual signals inefficiently because LLMs do not process the visual signals as a whole, instead they only focus on certain sub-areas whose tokens will be aggregated together in the decoder layers and further be processed to a higher level of abstraction that can be comprehended by the LLMs. As shown in Figure 1, after the initial layers of the decoder (specifically, layer 2), the LLM starts to focus on certain visual subareas, thus most of the visual tokens contribute little to LLM understanding while significantly slow down the inference process.

Existing works either i) follow a training-free paradigm that relies on attention scores or image-text relevance to adaptively select the most significant tokens [14], [15], [16], [17], or ii) adopt a tuning-based strategy that directly compresses visual tokens through pooling or learnable networks [18], [19], [17], [20], [21], equivalent to adding another abstraction level. On the one hand, training-free paradigm considers tokens with high attention score or relevance beneficial, discarding the rest tokens with small values, which fails to make full use of the discarded tokens. As a result, the details in the visual signals

Xin Wang, Zirui Pan, Hong Chen, Wenwu Zhu are with the Department of Computer Science and Technology, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China. E-mail: {xin_wang, wwzhu}@tsinghua.edu.cn, {pzr24, chen20}@mails.tsinghua.edu.cn. This is an invited paper. Corresponding Author: Wenwu Zhu. This work is supported by National Natural Science Foundation of China No.62222209, Beijing National Research Center for Information Science and Technology under Grant No.BNR2023TD03006.
Digital Object Identifier 10.1109/TCSVT.2025.3567138

are neglected, deteriorating their performance for fine-grained visual understanding tasks. On the other hand, tuning-based strategy increases training overhead, and further introduces information loss that reduces the generalization ability since similar visual signals (e.g., images and videos etc.) may become completely indistinguishable after compression.

To tackle the above issues, we propose to accelerate inference speed for LVLMs by adaptively compressing the visual tokens, which poses the following challenges.

1) Compression will inevitably bring information loss, and it is always difficult to achieve fast inference speed without performance drop.
2) Compressing visual tokens may result in low generalizability, since different visual signals may be shrunk into similar tokens, making it difficult to maintain good performances on new or unseen real-world datasets.

To address the challenges, we propose **Dis**entangled **Vi**sual Token **Co**mpression for Efficient Visual-Language Model, dubbed DiViCo. To the best of our knowledge, DiViCo is the first attempt to explore the potential of disentangled encoding of visual signals in LVLMs. DiViCo performs token compressions in the K^{th} decoder layer of the LLM model. Firstly, we reuse the self-attention matrix of tokens from the K-1^{th} layer without bringing additional computational cost. We then rank the visual tokens according to their average attention scores and select the top $\tau\%$ tokens as the most significant. We note that these selected tokens are only part of the visual signals. Moreover, we utilize the remaining $(1-\tau\%)$ tokens via employing a shallow neural network to predict the information gap between using only the selected tokens and using all the tokens including these remaining tokens in a disentangled and variational manner. Specifically, we sample the information gap from a Gaussian distribution and predict the corresponding mean and variance. We further adopt the Kullback-Leibler (KL) Divergence [22] loss to guarantee the disentanglement within the information gap, ensuring its informativeness. We incorporate the predicted information gap into the $\tau\%$ selected visual tokens to derive the final compressed visual tokens, obtaining a compression rate of $1-\tau\%$. In this way, we are able to significantly reduce the number of visual tokens without losing valuable information and details, as well as increase the generalizability of the LVLM via the disentanglement design. The contributions of this work can be summarized as follows.

- We propose a novel Disentangled Visual Token Compression module, i.e., DiViCo, to effectively compress the visual tokens and maintain good performance simultaneously. To the best of our knowledge, this is the first attempt to explore the potential of disentangled encoding for visual signals in LVLMs towards inference speed acceleration. Moreover, our DiViCo module can be easily plugged into most existing LVLMs.
- We adaptively select the $\tau\%$ most important visual tokens and compress the remaining tokens in a disentangled and variational way. Compared to existing state-of-the-art methods, the proposed DiViCo module is able to significantly reduce the inference cost for LVLMs with little performance drop.
- We conduct extensive experiments to show the superiority of our proposed DiViCo module against several state-of-the-art baseline models by validating its significant improvement over a wide range of benchmarks with many backbones.

II. Related Work

In this section, we review related works on Large Vision-Language Model, variational encoders and visual compression for LVLM, respectively.

A. Large Vision-Language Model

Large Vision-Language Models (LVLM) [18], [11], [10], [23], [6] have emerged as a cornerstone in multimodal artificial intelligence, enabling systems to process and understand both visual and textual information seamlessly. Pioneering architectures such as CLIP [24] have demonstrated the power of joint training on large-scale image-text datasets to create models capable of performing zero-shot tasks across various domains. These models leverage vision encoders [25] and text encoders to align visual and textual representations in a shared embedding space. Their scalability and generalization capabilities have established LVLMs as foundational technologies in applications ranging from image retrieval to caption generation.

Building on these foundations, subsequent works such as Flamingo [6], BLIP2 [23], and Qwen-VL [10] have incorporated cross-modal attention mechanisms to enhance interactions between modalities. These models are designed to process both image and text inputs simultaneously, allowing for improved performance on complex reasoning tasks such as Visual Question Answering (VQA) and multimodal dialog systems. The introduction of fine-tuning paradigms such as adapter-based methods, Prompt Learning [26], and Low-Rank Adaptation (LoRA) [27] has further improved the adaptability of LVLMs to downstream tasks with minimal computational overhead.

Despite these advancements, one of the critical challenges faced by LVLMs is their computation and memory inefficiency, especially when handling long visual tokens. Models such as Flamingo or LLaVA [11] employ dense cross-attention between image and text tokens, leading to quadratic complexity with respect to token counts. This limitation has motivated researchers to dive into token reduction strategies, such as visual token pooling, clustering, or employing learned compression modules, to decrease the token length without sacrificing critical information.

B. Variational Encoder

Variational encoders, or Variational Autoencoders (VAEs) [28], [29], have gained prominence as powerful tools for learning latent representations of data in an unsupervised or semi-supervised manner. VAEs integrate probabilistic reasoning with deep learning by modeling the latent space as a distribution rather than a fixed point. This approach enables the

generation of diverse and realistic samples, as well as robust representation learning for tasks such as reconstruction [30], [31], anomaly detection [32], and disentanglement [33], [34], [35], [36].

In the context of vision-language models, variational encoders have shown potential in addressing the complexity of high-dimensional data. By learning compact and structured latent representations of visual or textual information, variational encoders can significantly reduce token length while preserving essential features. For instance, recent works employ VAEs to encode visual features into a latent space [37], [38], [39], [40], [41], followed by reconstruction modules to ensure that the compressed representation retains sufficient fidelity for downstream tasks. This has paved the way for integrating VAEs into token compression pipelines in multimodal systems.

One of the critical strengths of variational encoders lies in their flexibility to incorporate domain-specific priors [42], [43]. Such conditioning mechanisms improve the informativeness of the latent variables, enabling more efficient compression and better generalization.

C. Visual Compression for Vision-Language Model

Visual compression has been first investigated for Vision-Language Models (VLMs), specifically for its vision transformers [44], [45], [46]. As in the era of large models, visual compression has become increasingly significant, since current high resolution images or videos will consume even larger memory space. Recently, visual compression, i.e., token reduction, for LVLMs can be categorized into two branches, namely training-free and tuning-based. For the former, FastV [14] ranks the visual tokens based on their attention scores from previous layers. Afterwards tokens with lower attention scores are discarded in the following decoder layers. ToMe [15] prunes the tokens based on the relevance between visual tokens and text and merges both modalities through the Bipartite Soft Matching algorithm [15]. These are the two pioneering works for token reduction. More recently, SparseVLM [16] makes use of the guidance from text tokens, and prune the visual tokens according to the relevance across decoder layers adaptively. Although it adopts a token recycling strategy to retrieve tokens from the deleting pool, tokens with less relevance still tend to be discarded with no doubt. For the latter, Deco [19] utilizes an average pooling layer to downsample the input visual signal at the patch level, and retrains the linear projector that adapts the compressed information to the LLM. LLaVA-Prumerge [17] samples the important visual tokens based on their relevance with the class tokens, which are then clustered via k-nearest neighbor around certain centers to get a compressed representation. Llama-vid [18] compresses the visual signals using average pooling and linear projectors. It further aggregates the compressed information with guidance from the text domain. We can see that the compression strategies they adopt may introduce much irreversible information loss, with many visual tokens directly discarded or indirectly compressed. Moreover, different visual signals, e.g., images or videos, can be indistinguishable under these compression strategies, since they are basically a projection from a high-dimensional space to a low-dimensional one, thus reducing the generalizability of the LVLM.

III. The proposed DiViCo Module

In this section, we first introduce some preliminaries, i.e., attention mechanism, KL divergence and basic knowledge on LVLM, then describe the specific implementation details of the proposed DiViCo module, including adaptive token selection and disentangled compression. We further provide the complete training procedure and the theoretical complexity analysis. The overall framework of DiViCo as well as its incorporation into LVLM are demonstrated in Figure 2.

A. Prelinminary

a) Attention Mechanism in LVLMs: The transformer layer in a typical LVLM adopts the casual self-attention design [48]. For the single-head implementation, an attention matrix $\mathbf{A} = \mathbf{Q} \times \mathbf{K}^T \in \mathbb{R}^{N \times N}$ is computed using $\mathbf{Q} \in \mathbb{R}^{N \times d}$, i.e., *Query*, and $\mathbf{K} \in \mathbb{R}^{N \times d}$, i.e., *Key*, where N and d represents the number and the dimension of all tokens, respectively. Afterwards the *Value* matrix $\mathbf{V} \in \mathbb{R}^{N \times d}$ will be aggregated using the corresponding weights in $\mathbf{A}$, formally,

$$\textit{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \textit{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}. \tag{1}$$

For self-attention, it takes the same embedding $\mathbf{E}$ as an input, then converts it to three matrices through linear projections and feeds them into an attention layer as follows,

$$\textit{Self-Attention}(\mathbf{E}) = \textit{Attention}(\mathbf{E}\mathbf{W}^Q, \mathbf{E}\mathbf{W}^K, \mathbf{E}\mathbf{W}^V), \tag{2}$$

where the projection matrices $\mathbf{W}^Q$, $\mathbf{W}^K$ and $\mathbf{W}^V \in \mathbb{R}^{d \times d}$ can make the attention mechanism more flexible. In DiViCo, we utilize the existing self-attention matrix to discover the most significant visual tokens, i.e., which receives the highest average attention scores.

b) KL Divergence Loss: The KL Divergence Loss [49], [50] is a measure in statistics that quantifies in bits the degree of closeness between a probability distribution $p = \{p(z_i)\}$ and a model distribution $q = \{q(z_i)\}$, formally,

$$\mathcal{D}_{KL}(q||p) = \sum_i p(z_i) \log \frac{q(z_i)}{p(z_i)}. \tag{3}$$

$\mathcal{D}_{KL}$ is non-negative, and $\mathcal{D}_{KL} = 0$ if and only if the two distributions are exactly identical. KL divergence loss is widely-used in variational autoencoders (VAEs). Particularly, if we choose a prior p that satisfies $p(z) = \prod_i p(z_i)$, penalizing the KL divergence will encourage the disentanglement across the dimensions (i.e., z_i) in z.

c) Large Vision-Language Model (LVLM): Large Vision-Language Models arise in recent years, and have achieved great success in complicated multimodal tasks. Current approaches usually adopt a *Pre-training*, *Fine-tuning*, *Predicting* paradigm [51], where an LVLM is pre-trained with large-scale image-text or video-text pairs, and further fine-tuned to learn the relevance between the visual signals and text with respect to specific tasks such as visual question answering. For a typical LVLM, the visual signal and text will be encoded into

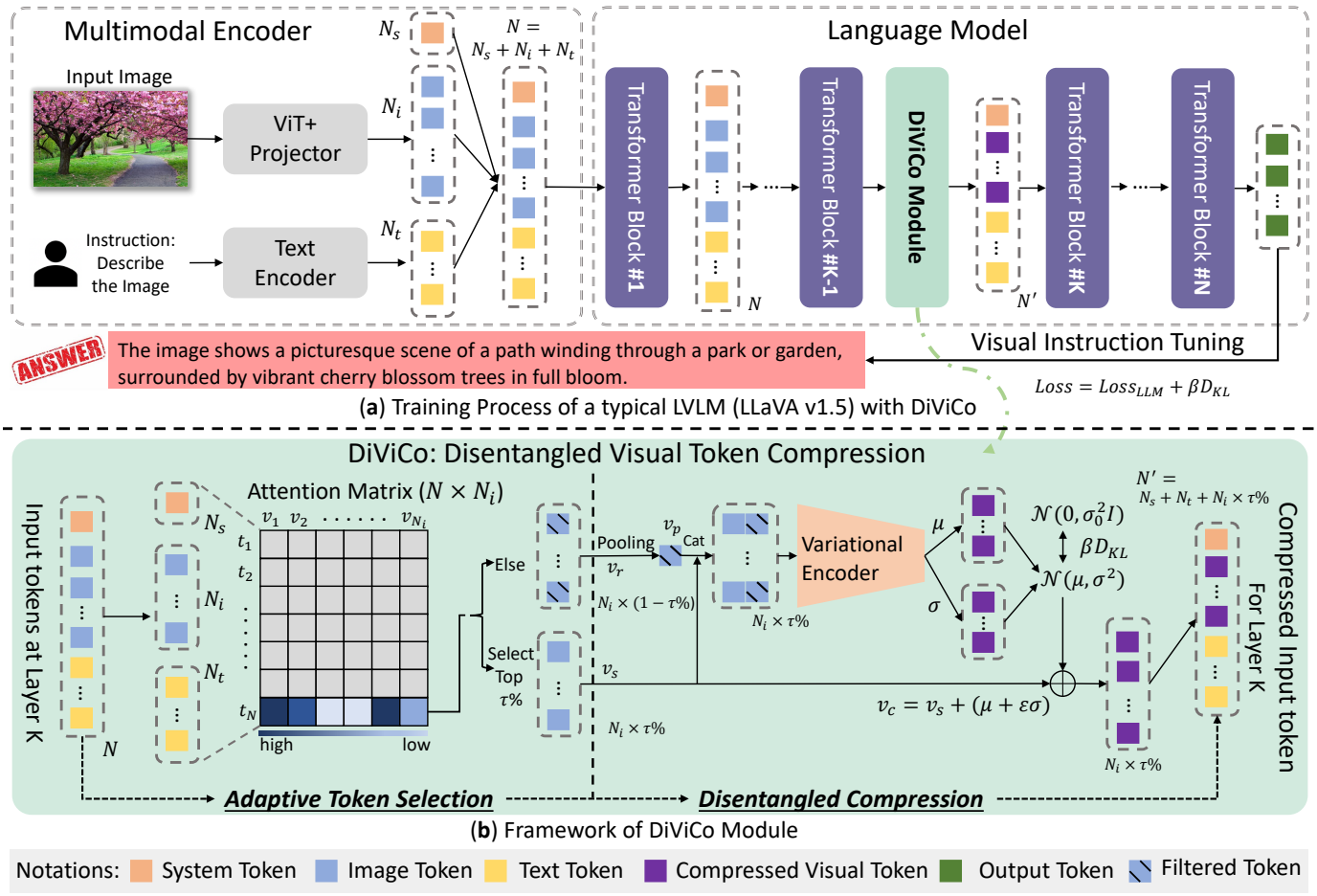


Fig. 2. The overall framework of an LVLM with the proposed DiViCo module. Subfigure (a) shows the training process of a typical LVLM, i.e., LLaVA-1.5 [11]. Typically, the images and texts are encoded separately before being concatenated and fed into the LLM, which usually adopts the decoder-only architecture [47]. Our proposed DiViCo Module is inserted between the $(K-1)^{th}$ and K^{th} layer of the decoder. Subfigure (b) demonstrates the detailed architecture of DiViCo. In concrete, we first perform adaptive token selection, where we retrieve the input visual tokens from layer K, and sample the top $\tau\%$ according to its average attention score. Next for the rest of the tokens, we employ a shallow neural network, i.e., variational encoder, to predict the information gap, i.e., performance differences between using only the selected tokens and using the whole original visual signals. Instead of directly predicting the information gap, we sample its value from a Gaussian distribution. In inference stage, the mean, i.e., μ, is adopted as the predicted result. We further introduce a KL divergence loss to keep the compressed information disentangled in the latent space. Then we perform the reconstruction with the information gap being incorporate into the selected visual tokens, which completes the procedure. These compressed visual tokens are utilized as the input to the following decoder layers, thus greatly relieving the computational burden during the inference process.

a set of tokens v_I and v_T respectively and be concatenated together to feed into the corresponding LLM, which usually employs a *decoder-only* architecture. The DiViCo module will be plugged in the decoder to relieve the computational overhead caused by the large number of visual tokens.

B. Adaptive Token Selection

We regard the degree of *attention* that a token is able to contribute (we may also call it *contributed attention*) within the *self-attention* mechanism of an LLM as the significance of this token. As such, higher contributed attention indicates more importance of the corresponding token in affecting the afterwards forward process. More specifically, we select the last token and compute the average attention weights it receives from any other visual token across all heads. However, as is shown in Figure 1, the average contributed attention of a token may shift from a scattered distribution to a more centralized one as the decoder layer goes deeper and deeper. Thus we can conclude that the LLMs gradually focus more on some anchor visual tokens, which inspires us to loose the grip on the rest of the visual tokens to boost efficiency.

Concretely, suppose all the tokens are denoted as a set $v = \{v_S, v_I, v_T\}$, where $v_S \in \mathbb{R}^{N_s \times d}$, $v_I \in \mathbb{R}^{N_i \times d}$ and $v_T \in \mathbb{R}^{N_t \times d}$ denote the set for system tokens, visual tokens and text tokens, and d represents the hidden size, while N_s, N_i and N_t are the corresponding number of tokens, respectively. We then make use of the existing *self-attention* matrix $\mathbf{A} \in \mathbb{R}^{H \times N \times N} = (a_{h,i,j})_{1 \le i,j \le N, 1 \le h \le H}$ that the LLM will compute in Layer K-1, so as not to bring any unnecessary computational cost, where $N = N_s + N_i + N_t$ represents the total number of tokens, and H stands for the number of heads in Multi-Head Attention. Then we compute the average attention score s_t for each visual token $v_i^{(t)} \in v_i$ as follows,

$$s_t = \frac{1}{H} \sum_{h=1}^{H} a_{h,-1,t}, \tag{4}$$

where $a_{h,-1,t}$ represents the attention score between the t^{th} visual token and the last token within the h^{th} head in Multi-Head Attention. We then select the visual tokens with the top $\tau\%$ average attention score as $v_s \in \mathbb{R}^{N_i \times d}$, i.e., $v_s = \{v_I^{(t)} | v_I^{(t)} \in v_I, s_t \textit{ ranks in top } \tau\%\}$, where $N_i^{'} = N_i \times \tau\%$. We note that the rest of the visual tokens with less average attention score, denoted as $v_r \in \mathbb{R}^{(N_i - N_i^{'}) \times d}$, will not be directly discarded. Given that v_r generally contains much less information, we perform an average pooling to obtain $v_p \in \mathbb{R}^d$ where $v_p = \textit{AveragePooling}(v_r)$, which is able to condense the information for the sake of efficiency.

C. Disentangled Compression

In section III-B, we derive the most significant visual tokens denoted as v_s, while the rest ones are denoted as v_r. We note that although v_s has incorporated most of the information, there still exists an information gap between v_s and the original visual signal. Ignorance of this gap may result in loss on some detailed information, thus deteriorating the model's capability of fine-grained understanding. Therefore, we propose to use disentangled encoding to predict the information gap based on the less significant tokens, i.e., v_r.

Specifically, for any chosen $v_s^{(t)} \in v_s$, we first concatenate it with v_p to derive a new token $v_s^{(t)'} \in \mathbb{R}^{2d}$ with dimension doubled from d to $2d$. We utilize a variational encoder $g_\theta(.) : \mathbb{R}^{2d} \rightarrow (\mathbb{R}^d, \mathbb{R}^d)$ which is a shallow neural network to model the posterior distribution $q(v_c^{(t)'} | v_s^{(t)'})$ after compression, where $v_c^{(t)'}$ represents the compressed information gap corresponding to $v_s^{(t)'}$, and θ stands for the learnable parameters. Finally we add $v_c^{(t)'}$ back to $v_s^{(t)}$ to derive the final compressed visual token $v_c^{(t)} = v_c^{(t)'} + v_s^{(t)}$.

We assume that $q(v_c^{(t)'} | v_s^{(t)'})$ follows an nC-dimensional multivariate Gaussian distribution. Thus we predict its mean μ and variance σ^2 using $g_\theta(.)$, and sample $v_c^{(t)'}$ from the Gaussian distribution. We use a simple re-parameterization [28] trick, introducing a random variable $\epsilon \sim \mathcal{N}(0, 1)$ to ensure the gradient will be back-propagated smoothly during training, since the modeled multivariate Gaussian distribution is intractable. During the inference stage, we use the predicted mean as $v_c^{(t)'}$. We note that modeling a distribution instead of a single value will enhance the robustness of the model, increasing its generalizability for unseen data. We argue that when the input visual signals, i.e., image or video, are disturbed by noise, our proposed model will still perform well (See Figure 5 and Table II). Since we concatenate $v_s^{(t)}$ and v_p as the input, as well as capture the information gap between $v_s^{(t)}$ and the original visual signal, $v_c^{(t)'}$ is expected to contain the information in v_p that is most relevant to $v_s^{(t)}$. Formally, the overall process can be summarized as follows,

$$\mu, \sigma \leftarrow g_\theta(v_s^{(t)'}), \tag{5}$$

$$q(v_c^{(t)'} | v_s^{(t)'}) \sim \mathcal{N}(\mu^{(1)}, \mu^{(2)}, \cdots, \mu^{(d)}, \sigma^{(1)}, \sigma^{(2)}, \cdots, \sigma^{(d)}), \tag{6}$$

$$v_c^{(t)'} \leftarrow \mu + \epsilon \cdot \sigma, \tag{7}$$

where $\mu = [\mu^{(1)} \cdots \mu^{(d)}]$ and $\sigma = [\sigma^{(1)} \cdots \sigma^{(d)}] \in \mathbb{R}^d$. We further propose a KL divergence loss to ensure the disentanglement of the compressed visual token, improving its informativeness,

$$\mathcal{D}_{KL} = \mathcal{D}_{KL}\left(q(v_c^{(t)'} | v_s^{(t)'}) \,||\, p(v_c^{(t)'})\right), \tag{8}$$

where $p(v_c^{(t)'})$ is a prior distribution that satisfies $p(v_c^{(t)'}) = \prod_{j=1}^{d} p(v_{c,j}^{(t)'})$, which is mutually independent across all dimensions. Therefore, via optimizing $\mathcal{D}_{KL}$, we are able to minimize its distance to the independent distribution $p(v_c^{(t)'})$, enhancing the disentanglement within the dimensions of $v_c^{(t)'}$, and then forcing the compressed visual token to contain as much information as possible. On the other hand, the original training objective of LVLM will learn to reconstruct the ground-truth text tokens, which is consistent with our goal to reconstruct the original visual signal from our compressed ones. Therefore, by adding $\mathcal{D}_{KL}$ to the original loss, we can ensure the compressed visual tokens disentangled in latent space and reconstructable with respect to the original visual signal simultaneously. In practice, we design $g_\theta(.)$ as a double-layer Multi-Layer Perceptron (MLP), and set $p(v_c^{(t)'})$ as an n-dimensional independent standard Gaussian distribution.

D. Implementation

a) Training Stage: We equip the LVLM with the DiViCo module, and further fine-tune the module via utilization of existing pre-trained LVLM weights. We remark that the DiViCo module is fine-tuned together with foundation LLM model as a whole instead of being trained separately. Otherwise, the LLM may not be able to understand the compression strategy proposed in DiViCo, thus failing to uncover useful information from the newly compressed visual tokens.

Specifically, for LLaVA-v1.5, we adopt the visual instruction tuning [11] technique, fine-tuning DiViCo module and the Vicuna LLM at the same time. We add LoRA layers to the decoder layer of LLM, and fine-tune these LoRA layers while keeping the weights of the original layers fixed. Furthermore, we add a hyper-parameter β to control the weights of the KL divergence loss D_{KL} as follows,

$$\texttt{loss} \leftarrow \texttt{loss}_{LLM} + \beta \cdot \mathcal{D}_{KL}. \tag{9}$$

The detailed training procedure can be found in Algorithm 1.

b) Inference Stage: During inference, we use the mode (mean) of $q(v_c^{(t)'} | v_s^{(t)'})$ as the predicted information gap between the selected significant visual tokens and the original visual signal. We control the compression rate by adjusting the hyper-parameter $\tau\% \in (0, 1)$. Larger τ indicates smaller compression rate, leading the model to achieve a relatively higher accuracy at the cost of lower efficiency, while smaller τ indicates larger compression rate, resulting in faster inference speed. The overall inference pipeline is illustrated in Figure 3, where the model is instructed to describe the main content of a given visual image. We demonstrate the responses from LLaVA-v1.5-13B equipped with DiViCo module with different compression rates (i.e., $\tau\% \in \{1, 0.16, 0.11\}$). From the results we can observe that LLaVA-v1.5-13B equipped with DiViCo performs fairly well at large compression rates, generating nearly identical answers compared to the non-compressed version which has a much slower inference speed.

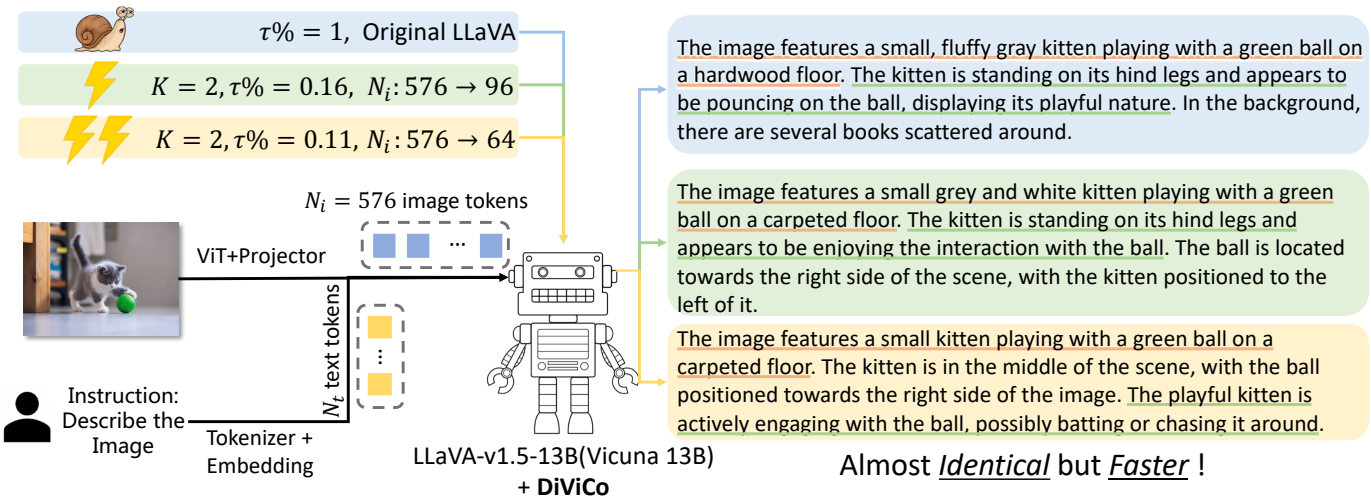


Fig. 3. The inference pipeline for LLaVA-v1.5-13B equipped with the proposed DiViCo module. Here, the user will input the visual signal, i.e. image, and the instruction to the LVLM. For original LLaVA, the image will be encoded into $N_i = 576$ tokens. With DiViCo, if we modify the hyper-parameter $\tau\%$ to 0.16% and 0.11%, the number of image tokens will be reduced to 96 and 64, respectively. The responses of the two that load DiViCo module with different hyper-parameter τ are demonstrated in green and yellow boxes, where the similar parts across all three boxes are marked with underline of the same color. From the results we can see that the responses outputted by LLaVA equipped with DiViCo are almost identical to the one outputted by the original LLaVA, while the former achieves a much faster speed.

Algorithm 1 Training procedure of DiViCo.

Input: K, τ, β, $\texttt{Data} = \{(\texttt{image}, \texttt{text})\}_{N_D}$
Parameter: $\theta = \{$parameters for DiViCo, parameters for LLM in terms of LoRA$\}$
function DISENTANGLEDENCODING($v_s^{'}$)
  $\mu, \sigma \leftarrow g_\theta(v_s^{'})$.
  $\epsilon \sim \mathcal{N}^{(d)}(0,1)$
  $v_c^{'} \leftarrow \mu + \epsilon\sigma$
  return $v_c^{'}$.
end function
BEGIN MAIN FUNCTION:
Initialize H, N, d as number of heads in multi-head attention, number of tokens and dimension of tokens, respectively.
repeat
  Forward a batch of data $\{(\texttt{image}, \texttt{text})\}_{N_{batch}}$.
  Select the input visual tokens v_I for layer K of the LLM decoder. Initialize N_i as the number of visual tokens.
  Make use of the existing self-attention matrix $\mathbf{A} \in \mathbb{R}^{H\times N\times N} = (a_{h,i,j})_{1\le i,j\le N, 1\le h\le H}$.
  for $t \leftarrow 0$ to N_i **do**
    Compute $s_t \leftarrow \frac{1}{H}\sum_h a_{h,-1,t}$
  end for
  $v_s \leftarrow \{v_I^{(t)} \mid v_I^{(t)} \in v_I, s_t \textit{ ranks in top } \tau\%\}$
  $v_r \leftarrow v_I \setminus v_s$
  $v_p \leftarrow$AVERAGEPOOLING(v_r).
  $v_s^{'} \leftarrow$CONCATENATE(v_s, v_p).
  $v_c^{'} \leftarrow$DISENTANGLEDENCODING($v_s^{'}$).
  $v_c \leftarrow v_s + v_c^{'}$.
  Update the input visual tokens for layer K of the LVLM decoder as v_c.
  Compute $\texttt{loss} = \texttt{loss}_{LLM} + \beta\mathcal{D}_{KL}$.
  Update θ.
until Converged

E. Complexity Analysis

In this section, we theoretically analyze the complexity of DiViCo module as well as the efficiency gains of LVLMs when equipped with DiViCo in terms of FLOPs (floating-point operations per second). For a typical Transformer layer [48], we consider that its computational overhead mainly comes from the operation performed in the Multi-head Attention layer (MHA) and the Feed-forward Network (FFN). Suppose the dimension of the intermediate layer of FFN is m, we can calculate the FLOPs for a single Transformer layer as follows,

$$4N_id^2 + 2N_i^2d + 2N_idm. \tag{10}$$

When equipped with DiViCo module, which is inserted before the K^{th} layer of the decoder, the number of visual tokens will be reduced from N_i to $N_i^{'}$, which equals to $N_i \times \tau\%$ (same proportions for the corresponding FLOPs). We aslo note that DiViCo module introduces some additional computational overhead, which is brought by the *Disentangled Compression*. Theoretically, the overall FLOPs reduced by DiViCo module can be calculated as follows,

$$\Delta\texttt{FLOPs} = (L-K)(4\bar{N}_id^2 + \frac{1+\tau\%}{1-\tau\%}\cdot 2\bar{N}_i^2d + 2\bar{N}_idm) - 4N_i^{'}d^2, \tag{11}$$

where L represents the total number of Transformer layers in LLM, and $\bar{N}_i = N_i - N_i^{'} = (1-\tau\%)N_i$. In practice, we select (K, τ) far less than $(L, 1)$, typically around $(2, 11.1\%)$ respectively, and the calculated $\Delta\texttt{FLOPs}$ can be approximated as follows,

$$\Delta\texttt{FLOPs} \xrightarrow{K\ll L, \tau\ll 1} L\cdot(4\bar{N}_id^2 + 2\bar{N}_i^2d + 2\bar{N}_idm). \tag{12}$$

Thus, the theoretical reduction ratio for FLOPs when equipped with DiViCo module can be approximated via the following equations,

$$\frac{4\bar{N}_id^2 + 2\bar{N}_i^2d + 2\bar{N}_idm}{4N_id^2 + 2N_i^2d + 2N_idm} \tag{13}$$

$$=(1-\tau\%)\cdot\frac{2d+m+\bar{N}_i}{2d+m+N_i} \tag{14}$$

$$\approx(1-\tau\%)\quad(\because N_i \ll 2d+m), \tag{15}$$

which approximately equals to $1-\tau\%$. Thus, we can safely use $1-\tau\%$ to represent the overall compression rate.

IV. Experiment

In this section, we empirically evaluate the performance of our proposed DiViCo module through quantitative experiments on real-world datasets against several state-of-the-art baselines. We also conduct a series of ablation studies, in which we test the efficacy of some core components in DiViCo, and study the effect of the hyper-parameter K on model performance. In addition, we provide some visualization experiments and efficiency analysis to further validate the effectiveness of our DiViCo module.

A. Experimental Setup

a) Datasets and Baselines: We conduct extensive quantitative experiments on several real-worlds datasets:

- TextVQA [52]: The TextVQA benchmark evaluates the model's reasoning abilities through challenging visual-answering tasks with rich textual information. It contains 45,336 questions on 28,408 images that require reasoning over texts to obtain the correct answer.
- Pope [53]: The Pope benchmark uses three sampling strategies to evaluate the degree of hallucinations in models via requiring it to answer a series of binary questions regarding the presence of objects in an image. Metrics such as *recall*, *precision* and *F1*, etc., will be calculated under each sampling strategy. In this work, we adopt the average *F1* metric as the final score.
- MMBench [54]: The MMBench benchmark evaluates the performance of models comprehensively across various and hierarchical dimensions, with the top level including *Perception* and *Recognition*, and the bottom level containing 20 specific ability dimensions.
- MME [55]: The MME benchmark also evaluates both the *Perception* and *Recognition* abilities of models comprehensively across many dimensions. It consists of 14 subtasks, each is carefully designed with concise instruction-answer pairs, thus effectively avoiding data leakage and enabling fair comparisons of LVLMs.
- VQAv2 [56]: The VQAv2 benchmark evaluates the visual perceptron abilities through open-ended questions. It consists of over 260,000 images, covering large quantities of real-world objects. Ten ground-truth answers are provided by human annotators for each question.
- Vizwiz [57]: The Vizwiz benchmark consists of over $31,000$ visual questions originating from a group of visually impaired people each of whom takes a picture using a mobile phone and records a spoken question about it, together with 10 crowdsourced answers per visual question.
- GQA [58]: The GQA benchmark consists of three parts, i.e., scene graphs, questions and images, evaluating the models' abilities to understand visual scenes. It develops a powerful and robust question engine that leverages the Visual Genome scene graph structures to create 22M diverse reasoning questions.
- TGIF [59]: The TGIF benchmark extends the image question-answering task to the video domain, consisting of over 160,000 video-question pairs, evaluating the models' abilities to comprehend details of the given videos.
- MSVD [60]: The MSVD benchmark is based on the existing Microsoft search Video Description dataset, which contains nearly 2,000 videos clips and corresponding question-answer pairs. Due to its large data size and question diversity, it is widely used in video question answering tasks. In addition, the questions are mainly formulated in five types, i.e., *What*, *Who*, *How*, *When* and *Where*.
- MSRVTT [60]: The MSRVTT benchmark proposes complicated understanding tasks, where models are required to effectively comprehend and reason over the videos in terms of both spatial and temporal information. Similar to MSVD, the questions in MSRVTT are also formulated in the five types.

We compare our methods with two most recent and powerful baselines, FastV [14] and DeCo [19], which are the representative works of two paradigms, i.e., training-free and tuning-based, respectively. Specifically, FastV ranks the visual tokens according to the calculated attention scores, while DeCo adds an extra average pooling layer to downsample the original visual tokens and learns a new projector to adapt the downsampled visual tokens to the LLM.

b) Implementation Details: In training, we set the hyper-parameters K and β to 2 and 0.02, respectively, and $\tau\% \in \{6.7\%, 11.1\%, 16.7\%\}$. The data we used to fine-tune both DiViCo and the corresponding LLM are consistent with LLaVA-v1.5 [11]. We train the models for 1 epoch on 4 Nvidia A100 80G GPUs. The inference phase follows the evaluation settings established by LLaVA-v1.5 [11], Qwen-VL [10] and Video-LLaVA [61], respectively. All the experiments are carried out on a single A100 80G GPU.

B. Main Results

In the main experiment, we equip the backbone models LLaVA-v1.5-7b[1], LLaVA-v1.5-13b[2] and Qwen-VL[3] with the proposed DiViCo module and baseline models (FastV and DeCo). We test the performances of these models at different compression rates, and the results are shown in Table I, where we additionally calculate the relative improvement of DiViCo over the best baseline.

From the results, we can observe that at different compression rates, i.e., 83.3%, 88.9% and 93.3%, the performance of DiViCo is consistently better than FastV and DeCo on various datasets. And generally, the larger the compression rate, the more superiorly that DiViCo competes against other baselines. Specifically, at compression rate of 93.3%, DiViCo achieves

[1]https://huggingface.co/liuhaotian/llava-v1.5-7b
[2]https://huggingface.co/liuhaotian/llava-v1.5-13b
[3]https://huggingface.co/Qwen/Qwen-VL-Chat

TABLE I
PERFORMANCE COMPARISON BETWEEN MULTIPLE BACKBONES EQUIPPED WITH DIVICO AND BASELINE METHODS.

Backbone	Compression Rate	Method	TextVQA(↑)	Pope(↑)	MMBench(↑)	MME(↑)
LLaVA-7b	-	Baseline	58.21%	73.73%	85.88%	1862.91
	$1-\tau\% = 83.4\%$ Token 576→96	FastV	55.16%	71.3%	56.7%	1689.61
		DeCo	45.10%	70.69%	62.39%	1384.86
		DiViCo	**55.25%**	**72.1%**	**63.46%**	**1769.73**
		Improvement	0.16%↑	1.12%↑	1.72%↑	4.74%↑
	$1-\tau\% = 88.9\%$ Token 576→64	FastV	55.30%	70.39%	59.63%	1564.05
		Deco	52.36%	64.77%	58.9%	1574.7
		DiViCo	**55.64%**	**71.03%**	**71.4%**	**1651.0**
		Improvement	0.64%↑	0.91%↑	19.74%↑	4.85%↑
	$1-\tau\% = 93.3\%$ Token 576→38	FastV	51.65%	67.1%	48.8%	1368.95
		DeCo	52.18%	54.92%	46.29%	1381.70
		DiViCo	**53.64%**	**68.7%**	**49.18%**	**1438.83**
		Improvement	2.80%↑	2.38%↑	0.78%↑	5.10%↑
		Overall Impro.	1.20%↑	1.47%↑	7.41%↑	4.90%↑
LLaVA-13b	-	Baseline	61.0%	76.5%	72.1%	1827.25
	$1-\tau\% = 83.3\%$ Token 576→96	FastV	58.19%	**74.02%**	62.0%	1747.35
		Deco	56.44%	71.3%	60.7%	1702.2
		DiViCo	**58.54%**	73.9%	**62.3%**	**1755.70**
		Improvement	0.60%↑	-0.16%↓	0.48%↑	0.48%↑
	$1-\tau\% = 88.9\%$ Token 576→64	FastV	56.09%	72.90%	54.7%	1668.36
		Deco	54.64%	70.69%	60.23%	1587.5
		DiViCo	**57.32%**	**73.3%**	**61.14%**	**1696.16**
		Improvement	2.19%↑	0.55%↑	1.51%↑	1.67%↑
	$1-\tau\% = 93.3\%$ Token 576→38	FastV	52.43%	68.8%	52.19%	1543.25
		DeCo	46.94%	58.96%	56.19%	1404.95
		DiViCo	**56.26%**	**71.7%**	**57.47%**	**1626.70**
		Improvement	7.30%↑	4.22%↑	2.28%↑	5.41%↑
		Overall Impro.	3.36%↑	1.54%↑	1.42%↑	2.52%↑
			TextVQA(↑)	Vizwiz(↑)	GQA(↑)	VQAV2(↑)
Qwen-VL	-	Baseline	61.33%	35.24%	58.0%	78.54%
	$1-\tau\% = 83.3\%$ Token 256→42	FastV	51.21%	31.28%	51.88%	53.74%
		DeCo	48.60%	30.94%	50.35%	53.19%
		DiViCo	**52.02%**	**32.20%**	**53.15%**	**56.47%**
		Improvement	1.58%↑	2.94%↑	2.45%↑	5.08%↑
	$1-\tau\% = 88.9\%$ Token 256→28	FastV	50.02%	30.03%	47.33%	44.3%
		DeCo	47.88%	30.78%	47.32%	44.01%
		DiViCo	**51.86%**	**31.45%**	**50.23%**	**47.7%**
		Improvement	3.68%↑	2.18%↑	6.13%↑	7.67%↑
	$1-\tau\% = 93.3\%$ Token 256→17	FastV	43.11%	24.12%	37.77%	36.00%
		DeCo	42.73%	26.80%	35.32%	37.03%
		DiViCo	**44.01%**	**28.43%**	**38.62%**	**40.46%**
		Improvement	2.09%↑	6.08%↑	2.25%↑	9.26%↑
		Overall Impro.	2.45%↑	3.73%↑	3.61%↑	7.34%↑

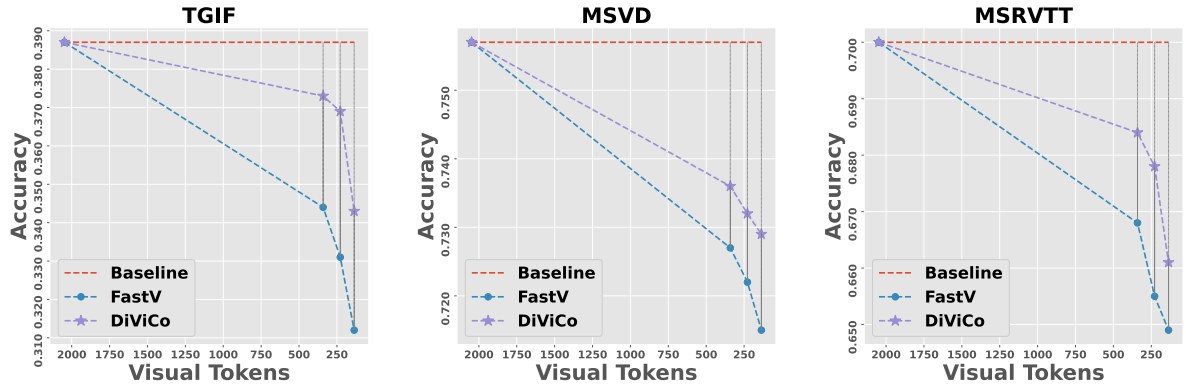


Fig. 4. Performance comparisons between DiViCo and FastV on Video-LLaVA with different compression rates and benchmarks.

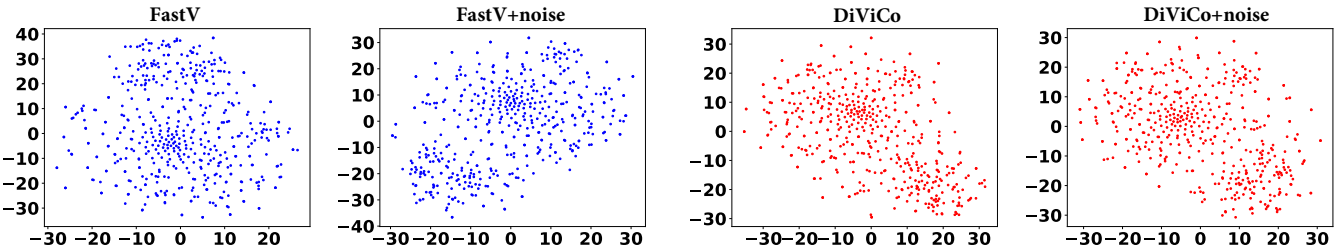


Fig. 5. Visualization of the distribution for the predicted next token embedding using t-SNE. We select the first 500 samples from dataset TextVQA for ease of demonstration.

over 7.3% gain of accuracy on TextVQA with LLaVA-13b, 6.08% gain of *F1* metric on Pope with Qwen-VL, and 5.10% gain of scores on MME with LLaVA-7b. These demonstrate the strong comprehension capabilities of our proposed DiViCo, as well as its generalizability. The performance boost may be attributed to the strategy proposed in DiViCo that we sufficiently utilize the less significant visual tokens, and adopt a disentangled compression approach to minimize the information loss.

C. Performance on the Video Domain

Our proposed DiViCo module can be easily implemented in LVLMs for video domain via treating videos as multiple frames. Therefore, we verify the effectiveness of DiViCo on three video understanding benchmarks. Specifically, we choose Video-LLaVA [61][4] as our backbone model, and compare the performance of DiViCo with FastV under different compression rates. Note that due to the spendy and irreducible identities of ChatGPT [3], we use LLaVA-v1.5-13b to assist the evaluation. The results are illustrated in Figure 4, where we mark the baseline that does not conduct any compression as a horizontal (red) line, indicating the upper bound of the performance. We can clearly observe that DiViCo outperforms FastV across all the datasets and compression rates, demonstrating its superiority under various compression scenarios for both images and videos.

D. Visualization

We conduct several visualization experiments to verify the effectiveness of our proposed method in terms of *Robustness*, *Disentanglement* and *Attention Score*.

[4]https://huggingface.co/LanguageBind/Video-LLaVA-7B

a) Robustness: DiViCo compresses the less significant visual tokens in a disentangled manner, thus it benefits in stronger generalizability towards unseen data. Additionally, DiViCo tends to be more robust when some random noises disturb the input data, since we model a distribution rather than a single point. To validate this, we select the first 500 samples in TextVQA, and equip LLaVA-v1.5-7b with DiViCo and FastV. At the compression rate of 88.9%, we add the same Gaussian noise to the input image data, and observe the changes of the distributions for the next predicted token after the disturbance. We use t-SNE [62] to reduce the dimension of the probability distribution and visualize it in Figure 5. From the results, we can observe that i) the distribution of the next token predicted by DiViCo stays almost unchanged after the disturbance, while ii) the distribution of the next token predicted by FastV changes drastically in contrast. Moreover, we quantitatively compare the performance of DiViCo against FastV under the same noise setting, whose results are demonstrated in Table II. The drop in performance for DiViCo after the noise disturbance is significantly less than the drop for FastV across all datasets. Specifically, the average performance drop for DiViCo is 2.315%, in comparison to the 5.19% drop (more than twice) for FastV, which further validates the robustness and generalizability of DiViCo.

b) Disentanglement: Additionally, we measure the disentanglement within the dimensions of v_c based on the independence level $\mathcal{IL}$ defined as follows,

$$\mathcal{IL} = 1 - \frac{2}{d(d-1)} \sum_{1 \le i,j \le d} |corr_{i,j}|, \tag{16}$$

where $corr_{i,j}$ is the correlation between the i^{th} and j^{th} dimension of v_c. Figure 7 shows the degree of disentanglement

TABLE II
PERFORMANCE COMPARISONS OF LLAVA-V1.5-13B EQUIPPED WITH DIVICO AND FASTV WHEN ADDING RANDOM GAUSSIAN NOISE. SPECIFICALLY, WE ADD THE SAME NOISE SAMPLED FROM A STANDARD GAUSSIAN DISTRIBUTION TO THE ENCODED IMAGES.

Method	TextVQA	Pope	MMBench	MME
FastV	56.09%	72.90%	54.7%	1668.36
FastV+noise	55.67%(↓ 0.75%)	70.7%(↓ 3.02%)	50.5%(↓ 7.68%)	1512.9(↓ 9.32%)
DiViCo	57.32%	73.3%	61.14%	1696.16
DiViCo+noise	57.10%(↓ **0.38%**)	72.58%(↓ **0.98%**)	58.7%(↓ **3.99%**)	1629.80%(↓ **3.91%**)

Fig. 6. Visualization of the selected significant visual tokens in the proposed DiViCo module. We show four examples from TextVQA dataset, where the original and compressed images with token attention are displayed from left to right, respectively.

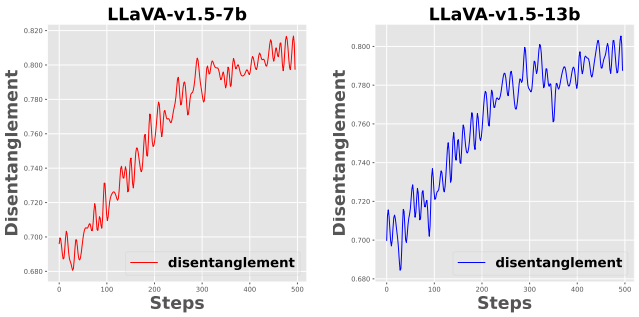


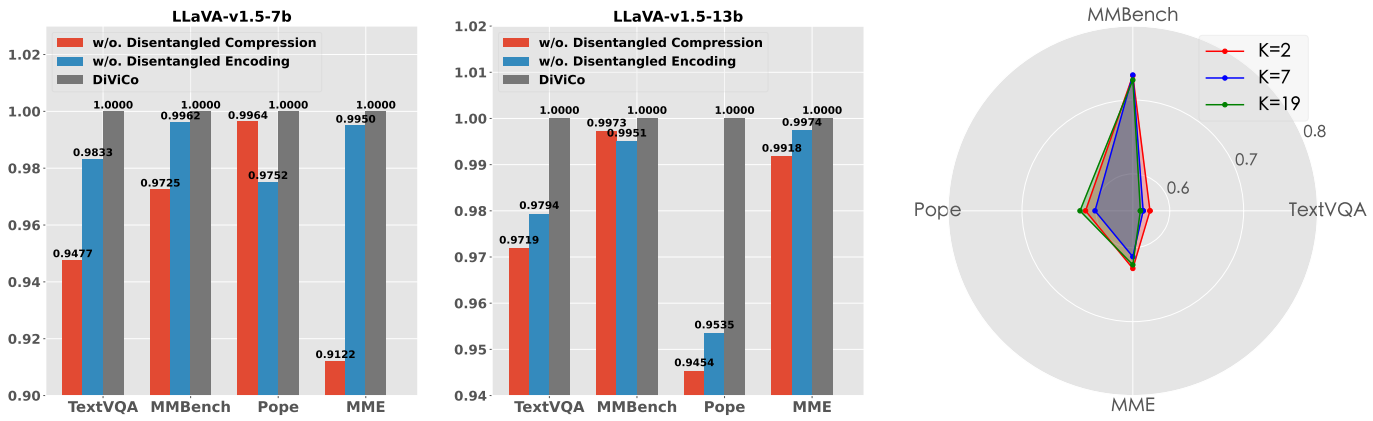
Fig. 7. Degree of disentanglement within compressed visual tokens of LLaVA-v1.5-7b and LLaVA-v1.5-13b equipped with DiViCo during different training steps.

within different training steps for LLaVA-v1.5-7b and LLaVA-v1.5-13b equipped with DiViCo. We observe that DiViCo is able to gradually reach a large degree of disentanglement during the training process, which may take credits from the KL divergence loss, i.e., minimizing the distance between the target distribution and the n-dimensional independent standard Gaussian distribution in the latent space. We argue that a large degree of disentanglement will result in high informativeness, which is particularly beneficial for LVLM compression.

c) Attention Score: We choose several examples from TextVQA, and visualize the selected visual tokens from DiViCo in Figure 6, with the original image and question on the left and bottom of each group, respectively. We follow the experimental settings in Section IV-A, and set $\tau\%$ to 11.1%. Despite the large compression rate, we observe that DiViCo successfully retains most of the essential visual information that will help the LVLM to correctly answer the question. Take the picture on the right side of the first row as an instance, DiViCo captures the most significant information, i.e., *Koala* on the changing table, which is exactly the correct answer to the question. Therefore, in this scenario, neglecting other visual tokens will do no harm to the model performance. However, it is impossible to capture all the vital information in most situations. Take the picture on the left side of the first row as an example, only part of the texts (birth) on the display are captured. Therefore, it is necessary to utilize those less important tokens so that no useful information is discarded. Moreover, we would also like to point out that our proposed disentangled encoding approach aims to compress the information gap between the selected $\tau\%$ significant visual tokens and the original visual signal, boosting both accuracy and efficiency of LVLMs.

E. Ablation Studies

We conduct several ablation studies to verify the effectiveness of each component in the proposed DiViCo module as



(a) (b) (c)

Fig. 8. Ablation study. Figure (a) and Figure (b) demonstrate the performance comparisons between DiViCo and its two variants. For *w/o. Disentangled Compression*, we remove the part for disentangled compression, only fine-tuning the LLM, and while for *w/o. Disentangled Encoding*, we remove the KL divergence and estimate a scalar point instead of the prior Gaussian distribution. All metrics are reported as ratios relative to the full DiViCo module. Figure (c) performs the sensitivity analysis on the hyper-parameter K, where K is chosen from $\{2, 7, 19\}$.

well as the reasonableness of the hyper-parameter selection.

a) Components of DiViCo: We train two variants of DiViCo, each excluding some components from the full DiViCo module. We quantitatively compare the performances of the two variants against DiViCo on the four datasets, i.e., TextVQA, Pope, MMBench and MME. The detailed descriptions of the two variants are as follows,

1) Variant-a, denoted as *w/o. Disentangled Compression*: In variant-a, we do not perform disentangled compression but conducting adaptive visual token selection alone, i.e., we discard those less significant visual tokens. Then we directly feed these selected visual tokens into LLM, and fine-tune LLM itself.
2) Variant-b, denoted as *w/o. Disentangled Encoding*: In variant-b, we change $g_\theta(.)$ from our proposed disentangled encoder to a vanilla encoder, directly processing the compressed visual tokens (estimate a point instead of a distribution) without the KL divergence loss. Then we fine-tune both the encoder and the LLM.

We use LLaVA-v1.5-7b and LLaVA-v1.5-13b as the backbone architecture respectively, and set $\tau\%$ to 11.1%. The results are demonstrated in Figure 8(a) and Figure 8(b). We can observe that compared to the full DiViCo module (the grey bar), both variants (the red bar and the blue bar) consistently perform worse on all the datasets with both the backbones. On the one hand, the accuracy of variant-a drops drastically without *Disentangle Compression*. On the other hand, adding the KL divergence loss indeed improve the accuracy by a noticeable margin. This ablation study validates our claims that it is necessary to make full use of the less significant tokens, and that the utilization of a disentangled encoder with KL divergence loss is beneficial.

b) Hyper-parameter K: We discuss the choice of the hyper-parameter K, which determines the layer within LLM decoder that we insert our DiViCo module. Specifically, we select K from $\{2, 7, 19\}$, and conduct experiments for backbone architecture LLaVA-v1.5-13b with the compression rate of 88.9% on all the four datasets, whose results are illustrated in Figure 8(c).

We observe that when $K > 2$, different Ks result in similar performances on the four datasets. Specifically, the three quadrilaterals that represent $K = 2$, 7 and 19 are nearly inseparable in Figure 8(c). The reason may be that in layer 2 of the decoder, the distribution of the average attention a visual token can contribute has shifted from a scattered distribution to a more centralized one, as is shown in Figure 1. As such, the rest of the tokens in layer 2 with small average attention scores carry a relatively small proportion of all noteworthy information hidden in the original visual signal. Therefore, filtering them out and compressing them in layer 2 are enough for the model to reach a certain compression rate while minimizing the information loss. For larger K, we empirically show that compressing the visual tokens at this layer will not significantly improve the accuracy. However, according to Equation 11, larger K will reduce the efficiency gain brought by the proposed DiViCo module. On the other hand, for $K \leq 2$, the model still processes the whole visual tokens, failing to correctly determine the significance of each visual token. As a result, filtering and compressing at this stage will result in large performance drops. Therefore, we choose $K = 2$ to maintain a reasonable trade-off, where we can achieve super efficiency gain (since $K = 2$ is still far smaller than the total number of layers for a typical LLM decoder) while maintaining relatively high accuracy.

F. Efficiency Analysis

We quantitatively analyze the efficiency gain brought by our proposed DiViCo module and compare it with two baseline methods, FastV and DeCo. We choose LLaVA-v1.5-7b as the backbone architecture, and conduct the efficiency experiment over TextVQA dataset. The results in terms of *Accuracy(↑)*, *FLOPs(↓)*, *GPU Memory(↓)* and *Cuda Time(↓)* are illustrated in Table III. The relative improvement of each method over the original LLaVA-v1.5-7b is demonstrated in the corresponding

TABLE III
EFFICIENCY ANALYSIS FOR LLAVA-V1.5-7B EQUIPPED WITH DIVICO AND BASELINE METHODS ON TEXTVQA. WE EVALUATE THE EFFICIENCY OF AN LVLM IN TERMS OF FOUR METRICS, I.E., *Accuracy*(↑), *FLOPs*(↓), *GPU Memory*(↓) AND *Cuda Time*(↓). ADDITIONALLY, WE PROVIDE THE RELATIVE IMPROVEMENT IN ROW Δ *Ratio*.

Model	-	Accuracy(↑)	FLOPs(T)(↓)	GPU Memory(GB)(↓)	Cuda Time(ms)(↓)
LLaVA-v1.5-7b	-	58.21%	4.995	15.66	325
LLaVA-v1.5-7b+**FastV**	1-τ%=83.3%	55.16%	2.171	13.11	283
	Δ ***Ratio***	5.23% ↓	56.5% ↓	16.2% ↓	12.9% ↓
LLaVA-v1.5-7b+**DeCo**	1-τ%=83.3%	54.10%	1.823	11.90	275
	Δ ***Ratio***	7.06% ↓	63.5% ↓	24.0% ↓	15.4% ↓
LLaVA-v1.5-7b+**DiViCo**	1-τ%=88.9%	**55.64%**	**1.612**	**10.95**	**157**
	Δ ***Ratio***	4.42% ↓	67.7% ↓	30.1% ↓	51.7% ↓
Overall Improvement	-	0.81% ↑	4.2% ↓	6.1% ↓	36.3% ↓

row Δ *Ratio*, and the overall improvement of DiViCo over the best performing baseline is shown in the bottom row.

We can observe from Table III that DiViCo achieves a better accuracy score at a larger compression rate compared to baseline methods, and meanwhile DiViCo costs the least FLOPs, GPU memory and Cuda time. Specifically, DiViCo is able to maintain 95.6% of the accuracy of the original model at the compression rate of 88.9%, surpassing 94.8% and 93.0% of the uncompressed accuracy obtained by FastV and DeCo at an even smaller compression rate of 83.3%. Furthermore, the overall improvement of DiViCo over the best baseline DeCo regarding *Cuda Time* is 36.3%. We argue that the improvement may take credits from the fact that DiViCo can effectively utilize all the rest visual tokens which receive small attention scores. Thus, we are able to include more information into the compressed visual tokens. Additionally, we note that LLaVA-v1.5-7b equipped with DiViCo significantly reduces 67.7% of the original FLOPs, 30.1% of the original GPU memory and 51.7% of the original Cuda time while maintaining 95.6% of the original accuracy. Similar results hold on other backbones.

V. DISCUSSIONS

Although DiViCo is designed to fully capture the information from the less significant visual tokens, it may not be as effective at small compression rates compared to itself at large compression rates. This is mainly due to the reason that at small compression rates, the number of discarded visual tokens is so small that they can hardly carry any useful information. Therefore, compressing these tokens may seldomly be beneficial for the LVLMs. In this case, the compression operation itself brings additional computational overhead and the compressed information may interfere the decision of LVLMs since it may deviate the model from the main noteworthy objects. The solution is to stop compressing those less significant tokens because the selected $\tau\%$ significant tokens alone are enough for the inference of the LVLM at small compression rates such as 10% to 20%. However, we also note that small compression rates can hardly help the improvement of efficiency, while our proposed DiViCo module can keep fairly high accuracy at very large compression rates.

VI. LIMITATION

One possible limitation for DiViCo, and also for most of the tuning-based compression methods, is that we may need to retrain our model, including the variational encoder and LoRA version of LLM, at every given compression rate. A feasible solution may be that we copy the disentangled variational encoder three times for different purposes, i.e., i) large compression rates, ii) medium compression rates and iii) small compression rates. Then a Mixture-of-Experts [63] mechanism will be adopted to mix the outputs of the three variational encoders for the LLM. Ideally, no matter what compression rate is chosen, the corresponding variational encoder will lead a major role. Since this work mainly focuses on the compression strategies for visual tokens, we leave it for investigation in future work.

VII. CONCLUSION

Current large vision-language models usually employ large quantities of visual tokens, most of which contribute little to the final performances, while significantly increasing the computational overhead. Existing works mainly remove the less important visual tokens, or insert trainable layers to directly compress the visual tokens, resulting in lost of much useful information, and deteriorating the ability of generalization. In this paper, we propose a novel DiViCo module, which first selects the most significant visual tokens based on its average attention scores, and then compresses the information hidden in remaining tokens with a disentangled and variational paradigm. DiViCo is able to largely reduce the number of visual tokens while maintaining the performances of the LVLMs at a fairly high level. We conduct extensive experiments including ablation studies and visualizations for many backbones on various real-world datasets against several state-of-the-art baselines to verify the effectiveness of DiViCo.

REFERENCES

[1] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, pp. 9, 2019.

[2] Tom B Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[3] 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.

[4] 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.

[5] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al., "Deepseek llm: Scaling open-source language models with longtermism," *arXiv preprint arXiv:2401.02954*, 2024.

[6] 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*, vol. 35, pp. 23716–23736, 2022.

[7] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al., "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 24185–24198.

[8] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia, "Mini-gemini: Mining the potential of multi-modality vision language models," *arXiv preprint arXiv:2403.18814*, 2024.

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

[10] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou, "Qwen-vl: A frontier large vision-language model with versatile abilities," *arXiv preprint arXiv:2308.12966*, 2023.

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

[12] 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*, 2024, pp. 26296–26306.

[13] David Marr, *Vision: A computational investigation into the human representation and processing of visual information*, MIT press, 2010.

[14] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang, "An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 19–35.

[15] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman, "Token merging: Your vit but faster," *arXiv preprint arXiv:2210.09461*, 2022.

[16] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al., "Sparsevlm: Visual token sparsification for efficient vision-language model inference," *arXiv preprint arXiv:2410.04417*, 2024.

[17] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan, "Llava-prumerge: Adaptive token reduction for efficient large multimodal models," *arXiv preprint arXiv:2403.15388*, 2024.

[18] Yanwei Li, Chengyao Wang, and Jiaya Jia, "Llama-vid: An image is worth 2 tokens in large language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 323–340.

[19] Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou, "Deco: Decoupling token compression from semantic abstraction in multimodal large language models," *arXiv preprint arXiv:2405.20985*, 2024.

[20] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi, "Instructblip: towards general-purpose vision-language models with instruction tuning," 2024.

[21] Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, Ying Shan, and Yansong Tang, "Voco-llama: Towards vision compression with large language models," *arXiv preprint arXiv:2406.12275*, 2024.

[22] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*, Wiley-Interscience, July 2006.

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

[24] 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*. PMLR, 2021, pp. 8748–8763.

[25] Alexey Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[26] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 139–149.

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

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

[29] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling, "Semi-supervised learning with deep generative models," *Advances in neural information processing systems*, vol. 27, 2014.

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

[31] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas, "Videogpt: Video generation using vq-vae and transformers," *arXiv preprint arXiv:2104.10157*, 2021.

[32] Shuyu Lin, Ronald Clark, Robert Birke, Sandro Schönborn, Niki Trigoni, and Stephen Roberts, "Anomaly detection for time series using vae-lstm hybrid model," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Ieee, 2020, pp. 4322–4326.

[33] Xin Wang, Hong Chen, Zihao Wu, Wenwu Zhu, et al., "Disentangled representation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[34] Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu, "Disentangled representation learning for recommendation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 408–424, 2022.

[35] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu, "Learning disentangled representations for recommendation," *Advances in neural information processing systems*, vol. 32, 2019.

[36] Xin Wang, Zirui Pan, Yuwei Zhou, Hong Chen, Chendi Ge, and Wenwu Zhu, "Curriculum co-disentangled representation learning across multiple environments for social recommendation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 36174–36192.

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

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

[39] Xintao Duan, Jingjing Liu, and En Zhang, "Efficient image encryption and compression based on a vae generative model," *Journal of Real-Time Image Processing*, vol. 16, pp. 765–773, 2019.

[40] Zhihao Duan, Ming Lu, Zhan Ma, and Fengqing Zhu, "Lossy image compression with quantized hierarchical vaes," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 198–207.

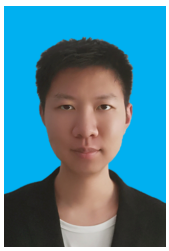
[41] Zhihao Duan, Ming Lu, Jack Ma, Yuning Huang, Zhan Ma, and Fengqing Zhu, "Qarv: Quantization-aware resnet vae for lossy image compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[42] Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Ler-

chner, "beta-vae: Learning basic visual concepts with a constrained variational framework.," *ICLR (Poster)*, vol. 3, 2017.
[43] Bin Dai and David Wipf, "Diagnosing and enhancing vae models," *arXiv preprint arXiv:1903.05789*, 2019.
[44] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie, "Not all patches are what you need: Expediting vision transformers via token reorganizations," *arXiv preprint arXiv:2202.07800*, 2022.
[45] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Mengshu Sun, Wei Niu, Xuan Shen, Geng Yuan, Bin Ren, Minghai Qin, et al., "Spvit: enabling faster vision transformers via soft token pruning (2022)," *URL https://arxiv. org/abs/2112.13890*.
[46] Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi, "Pumer: Pruning and merging tokens for efficient vision language models," *arXiv preprint arXiv:2305.17530*, 2023.
[47] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian, "A comprehensive overview of large language models," *arXiv preprint arXiv:2307.06435*, 2023.
[48] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017.
[49] MTCAJ Thomas and A Thomas Joy, *Elements of information theory*, Wiley-Interscience, 2006.
[50] Jonathon Shlens, "Notes on kullback-leibler divergence and likelihood," *arXiv preprint arXiv:1404.2000*, 2014.
[51] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu, "Vision-language models for vision tasks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
[52] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach, "Towards vqa models that can read," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8317–8326.
[53] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen, "Evaluating object hallucination in large vision-language models," *arXiv preprint arXiv:2305.10355*, 2023.
[54] 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?," in *European Conference on Computer Vision*. Springer, 2025, pp. 216–233.
[55] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al., "Mme: A comprehensive evaluation benchmark for multimodal large language models," *arXiv preprint arXiv:2306.13394*, 2023.
[56] 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*, 2017, pp. 6904–6913.
[57] 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*, 2018, pp. 3608–3617.
[58] Drew A Hudson and Christopher D Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6700–6709.
[59] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim, "Tgif-qa: Toward spatio-temporal reasoning in visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2758–2766.
[60] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang, "Video question answering via gradually refined attention over appearance and motion," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1645–1653.
[61] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan, "Video-llava: Learning united visual representation by alignment before projection," *arXiv preprint arXiv:2311.10122*, 2023.
[62] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.
[63] Saeed Masoudnia and Reza Ebrahimpour, "Mixture of experts: a literature survey," *Artificial Intelligence Review*, vol. 42, pp. 275–293, 2014.

Xin Wang is currently an Associate Professor at the Department of Computer Science and Technology, Tsinghua University. He got both of his Ph.D. and B.E degrees in Computer Science and Technology from Zhejiang University, China. He also holds a Ph.D. degree in Computing Science from Simon Fraser University, Canada. His research interests include multimedia intelligence, machine learning and its applications. He has published over 200 high-quality research papers in ICML, NeurIPS, IEEE TPAMI, IEEE TKDE, ACM KDD, WWW, ACM SIGIR, ACM Multimedia etc., winning three best paper awards including ACM Multimedia Asia. He is the recipient of ACM China Rising Star Award, IEEE TCMC Rising Star Award and DAMO Academy Young Fellow.

Zirui Pan is currently a Ph.D. student at the Department of Computer Science and Technology, Tsinghua University. He received his B.E. degree from the Department of Computer Science and Technology, Tsinghua University. His main research interests include curriculum learning, disentangled representation learning, and multi-modal generative AI.



Hong Chen received B.E. from the Department of Electronic Engineering, Tsinghua University, Beijing, China in 2020. He is currently a Ph.D. candidate in the Department of Computer Science and Technology of Tsinghua University. His main research interests include machine learning, multi-modal information processing.

Wenwu Zhu is currently a Professor in the Department of Computer Science and Technology at Tsinghua University. He also serves as the Vice Dean of National Research Center for Information Science and Technology, and the Vice Director of Tsinghua Center for Big Data. Prior to his current post, he was a Senior Researcher and Research Manager at Microsoft Research Asia. He was the Chief Scientist and Director at Intel Research China from 2004 to 2008. He worked at Bell Labs, New Jersey as Member of Technical Staff during 1996-1999. He received his Ph.D. degree from New York University in 1996. His research interests are in the area of data-driven multimedia networking and Cross-media big data computing. He has published over 400 referred papers and is the inventor or co-inventor of over 100 patents. He received eight Best Paper Awards, including ACM Multimedia 2012 and IEEE Transactions on Circuits and Systems for Video Technology in 2001 and 2019.

He served as EiC for IEEE Transactions on Multimedia (2017-2019) and IEEE Transactions on Circuits and Systems for Video Technology (2024-2025). He served in the steering committee for IEEE Transactions on Multimedia (2015-2016) and IEEE Transactions on Mobile Computing (2007-2010), respectively. He serves as General Co-Chair for ACM Multimedia 2018 and ACM CIKM 2019, respectively. He is an AAAS Fellow, IEEE Fellow, SPIE Fellow, and a member of The Academy of Europe (Academia Europaea).