Unified Lexical Representation for Interpretable Visual-Language Alignment

Yifan Li[1]* **Yikai Wang**[1]† **Yanwei Fu**[1]‡ **Dongyu Ru**[2]
Zheng Zhang[2] **Tong He**[2]†
[1]Fudan University [2] Amazon Web Services
yifanli23@m.fudan.edu.cn, yi-kai.wang@outlook.com,
yanweifu@fudan.edu.cn, {rudongyu, zhaz, htong}@amazon.com

Abstract

Visual-Language Alignment (VLA) has gained a lot of attention since CLIP's groundbreaking work. Although CLIP performs well, the typical direct latent feature alignment lacks clarity in its representation and similarity scores. On the other hand, lexical representation, a vector whose element represents the similarity between the sample and a word from the vocabulary, is a natural sparse representation and interpretable, providing exact matches for individual words. However, lexical representations are difficult to learn due to no ground-truth supervision and false-discovery issues, and thus requires complex design to train effectively. In this paper, we introduce LexVLA, a more interpretable VLA framework by learning a unified lexical representation for both modalities without complex design. We use DINOv2 as our visual model for its local-inclined features and Llama 2, a generative language model, to leverage its in-context lexical prediction ability. To avoid the false discovery, we propose an overuse penalty to refrain the lexical representation from falsely frequently activating meaningless words. We demonstrate that these two pre-trained uni-modal models can be well-aligned by fine-tuning on the modest multi-modal dataset and avoid intricate training configurations. On cross-modal retrieval benchmarks, LexVLA, trained on the CC-12M multi-modal dataset, outperforms baselines fine-tuned on larger datasets (e.g., YFCC15M) and those trained from scratch on even bigger datasets (e.g., 1.1B data, including CC-12M). We conduct extensive experiments to analyze LexVLA. Codes are available at https://github.com/Clementine24/LexVLA.

1 Introduction

The development of Vision-Language Alignment (VLA) models has made great progress [43, 48, 22, 26, 6] since the innovative CLIP [34] effectively learns a shared latent space where text and image are well-aligned. This progress has also boosted related fields like vision-language foundation models [20], multi-modal understanding [18], and text-conditional generation [35]. However, CLIP's latent features pose challenges of interpretability issue for analyzing individual factors' impact. Additionally, CLIP's visual model struggles to learn patch-level features, and its text model are trained based on incomplete and biased captions. These challenges reduces its overall effectiveness.

*Work completed during internship at AWS Shanghai AI Lab.

†Corresponding authors.

‡Dr. Fu is with School of Data Science in Fudan, Fudan ISTBI-ZJNU Algorithm Center for Brain-inspired Intelligence, Shanghai Key Lab of Intelligent Information Processing, and Technology Innovation Center of Calligraphy and Painting Digital Generation. Ministry of Culture and Tourism. China.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-0035

Figure 1: LexVLA can generate a lexical representation of the input image (the first word cloud figure), or to pick some patches of the image for local lexical information representation (the second word cloud figure, with the seleted patches boxed in red), and to select the most relevant patches of the image given the text content (the rightmost figure, with caption 'horse', with the second-to-last figure is the ground-truth mask).

On the other hand, the lexical representation is known for its clarity as each dimension corresponds to the similarity between the input and a specific word/token from the vocabulary. Additionally, lexical representation can naturally be made sparse, which means it's efficient for tasks like search or indexing in large-scale retrieval tasks. However, learning lexical representation is difficult. The embedding vector is much larger than the CLIP latent vector as the vocabulary size is usually much larger than CLIP's feature dimension. This poses several challenges: the lack of precise supervision signals; the incomplete and biased text supervision, and the high-dimensional embedding space with low-dimensional targets. Thus using lexical representations in CLIP-style VLA models is tough.

Previous lexical representation learning approaches tries to utilize strategies including ℓ_1 regularization [46, 3] to direct encourage sparisity or use ReLU activation [27] to remove negative scores [12, 5, 25] or directly regularize the floating-point operations [31]. However, these intricate training configurations usually introduce extra training difficulty and additional regularization such as log saturation [12, 25] and bag-of-words penalization [48] which potentially limits the VLA capacity.

In this paper, we propose LexVLA, a simple yet comprehensive framework for learning a unified lexical representation in the CLIP-style contrastive training pipeline, facilitating the VLA. LexVLA enjoys the following merits in addressing the challenges above. **1)** We use single-modal pre-trained models for their unique benefits, including DINOv2 [30] as vision model for its local-inclined features, and Llama 2 [40] as text model to learn lexical representations through in-context prediction tasks instead of traditional caption embeddings. **2)** We suggest a unified *vocabulary* with distinctive *codebooks* for each modality to maintain the strengths of single-modal pre-trained models by refrain them from learning the same feature space. This allows us to create better VLA models with less multi-modal training data and simpler setups. We also introduce an overuse penalty to prevent excessive activation of irrelevant tokens, improving upon the FLOPs loss [31] which only promotes sparsity. **3)** We incrementally train LexVLA to stay aligned with pre-trained models. For text, we fine-tune Llama 2 with LoRA adapter[14] and keep its codebook frozen. For vision, we freeze DINOv2 and train a projector with a self-attention layer and two multi-layer perceptions to the vision codebook, initialized using the text codebook. **4)** We introduce PatchDis, a metric for patch-level alignment in visual features, evaluating patch-level classification tasks with image patch features and category text tokens learned by VLA models. We propose PatchDis as a patch-level interpretability metric for VLA models not trained on fine-grained tasks like segmentation or detection.

Formally, our LexVLA employs a unified lexical vocabulary with distinct codebooks for text and image modalities, utilizing bi-encoders to individually encode inputs into lexical representations. These encoders, initially set with single-modal pre-trained models, undergo incremental fine-tuning via the standard contrastive learning objective, with the integration of an overuse penalty to promote sparsity and mitigate token over-activation. Emprically, we show the effectiveness of LexVLA on two zero-shot cross-modal retrieval benchmarks. Trained on the CC-12M dataset, it outperforms baselines fine-tuned on larger datasets like YF100m-CLIP + CC-12M and those trained from scratch on even larger datasets totaling 1.1B data, including CC-12M. We provide detailed analysis of LexVLA.

Our contributions can be listed as follows:

- We emphasize utilizing single-modal pre-trained models for vision-language alignment tasks to benefit from their unique properties that cannot learned by contrastive objectives. To exemplify, we use DINOv2 for its local-inclined features and Llama 2 for its in-context capacity.

https://doi.org/10.52202/079017-0035

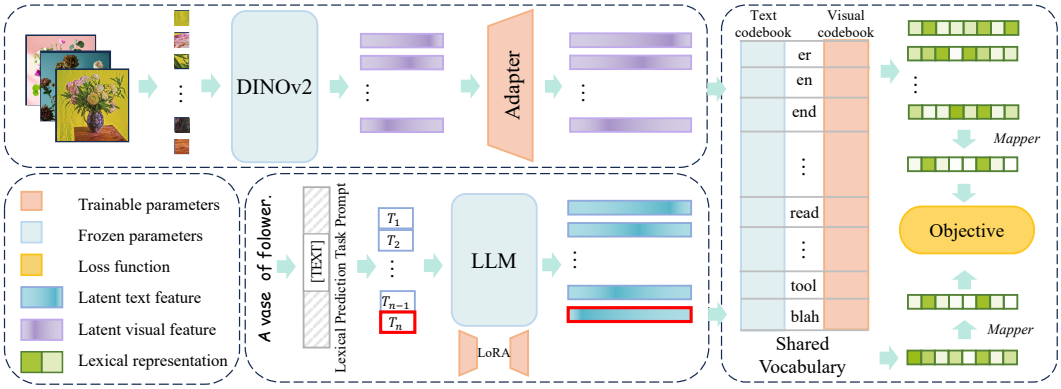

Figure 2: Framework of LexVLA. We learn a unified lexical representation with distinct codebooks for text and visual modalities. For the image, we adopt the frozen DINOv2, learn an adapter and a mapper to get visual lexical representation. For the text, we use LoRA to fine-tune the Llama 2 on in-context lexical prediction task, followed with a mapper to get the text lexical representation. We initialize codebooks as Llama 2's codebook, freeze the text codebook while fine-tuning the visual codebook. We train LexVLA with the standard contrastive objectives along with the proposed overuse penalty to encourage sparsity while preventing meaningless activation.

- We effectively learn a unified lexical representation with unique codebooks for vision and language modalities to refrain from the weakening of pre-trained capabilities.
- We propose an overuse penalty to encourage sparse embedding and prevent meaningless activation.
- We enjoy superior retrieval performance with less multi-modal training data, and achieve better patch-level interpretable VLA model with global supervision signals, quantitatively surpassing CLIP-style and lexical methods with our proposed patch-level interpretability metric PatchDis.

2 Related works

Vision-Language Alignment VLA has been a challenging problem in deep learning. The innovation of CLIP [34] has inspired a plethora of research efforts in combining visual and language modalities [21, 44, 26, 37, 6, 43]. With the rise of large language models (LLMs) [1, 40, 38], it's become common to use CLIP to get visual data from images and combine it with pretrained LLMs[24, 8, 20, 2, 49]. However, recent studies have identified visual encoders as potential bottlenecks in these models, posing issues like identifying actions and spatial relationships [41, 15], ignoring attributes and states [45, 10], introducing hallucination problems [39], etc. Nevertheless, progress in using other visual models instead of CLIP has been slow. Tong et al.[39] discovered a drop in instruction-following ability when integrating DINOv2[30] into LLaVA's [24] structure. To address this, they combined features from CLIP and DINOv2. Similarly, Zhai et al. [47] used a locked DINO for contrastive tuning, which necessitated significant aligned data and computational resources. In contrast, we use a locked DINOv2 for lexical VLA, demonstrating the superiority over CLIP with smaller amount of multi-modal training data. It is important to note that our work differs significantly from recent zero-shot cross-modal retrieval models, such as BEiT-3 [42]. Our approach emphasizes developing specific lexical representations for images and text, whereas general retrieval models lack this interpretable representation.

Multi-modal lexical representation Lexical representation is popular in information retrieval [12, 13] thanks to its interpretability and efficiency. However, adapting this approach to vision-language models is challenging because of the differences between the two modalities. Previous approaches rely on complex training stages (two [25] or three [5]), large scale multi-modal training data [5] to train effectively and additional Bag-of-Words (Bow) restrictions [25, 48] to prevent semantic misalignment. In contrast, we in this paper propose a single fine-tuning stage to align uni-modal pre-trained models in the output lexical space with less multi-modal training data and without BoW restriction. In this paper, our model shall be equipped with the generalizable ability for zero-shot cross-modal evaluation, rather than fine-tuning on each specific target cross-modal training dataset.

https://doi.org/10.52202/079017-0035

Additionally, it is important to note that lexical representation differs fundamentally from codebook strategies, as highlighted in [11]. While our aim is to develop interpretable lexical representations for images and visual patches at the vocabulary level, offering greater flexibility, codebook strategies focus on creating a unified yet non-explicit semantic code for images and text at the codebook level, which lacks the desirable traits of interpretability and high retrieval efficiency.

3 LexVLA

Problem setup Our target is to learn the lexical alignment between text and image modalities. Typically, the lexical representation $s_i \in \mathbb{R}^V, i \in \{\text{img, txt}\}$ is a score vector, whose element indicates the similarity between the sample and the corresponding word from the vocabulary $\mathbf{V}$ with $V = |\mathbf{V}|$. The contrastive lexical alignment aims to train the lexical encoders for each modality such that the similarity $\langle s_{img}, s_{txt} \rangle$ for positive image-text pairs are maximized and that for negative pairs are minimized, while preserving the correct lexical correspondence with the vocabulary.

Overview Our model, as in Fig. 2, learns a single lexical representation for VLA (Sec. 3.1). We encode each modality separately using uni-modal encoders (Sec. 3.2) adapted in Sec. 3.3.

3.1 Lexical representation

Vocabulary Inspired by tokenization techniques [36, 19], we initialize with the tokenizer vocabulary and refine it by removing unused and meaningless tokens, reducing the size from 32,000 to 17,149.

Codebooks The one-hot embedding is semantic-less and assume the same distance between different words, which is obviously not a good embedding. In this paper, we use the output codebook of language models as a well pre-trained and semantic-rich codebook, inducing lexical codes of 4096-dim vectors. However, the text codebook is not optimized for visual features. Basically, during the pre-training of language model, the visual world is not observable, thus we may have a gap between text-only embeddings and the visual-embeddings. To this end, we propose to use the same *vocabulary* $\boldsymbol{V}$ but a unique *codebook* $\boldsymbol{Z}_i, i \in \{\text{img, txt}\}$ for text and image modalities, respectively. As the language codebook is pre-trained on large corpus, we freeze $\boldsymbol{Z}_{txt}$ to enjoy the well-trained text embeddings. $\boldsymbol{Z}_{img}$ is initialized with the weights of $\boldsymbol{Z}_{txt}$ and fine-tuned along with the training of lexical encoders to enable the adaptation for visual features.

Sparse representation Thanks to the inherent similarity measurement in lexical representations, it is natural to turn a dense output vector into a sparse lexical representation. Typically we could utilize the following strategies: 1) Top-k thresholding: select the most important k items and discard the rest; 2) Value thresholding: keep items with value above the threshold and discard the rest. In this paper, we use the value thresholding strategy. We select items with values greater than $1/\sqrt{V}$, which means we consider only the items with above-average signals as informative.

3.2 Lexical encoder

Given the image as $\boldsymbol{x}_{img}$ and text as $\boldsymbol{x}_{txt}$, the lexical representation is achieved by:

$$\boldsymbol{s}_i := e_i(\boldsymbol{x}_i) = h_i \circ g_i \circ f_i(\boldsymbol{x}_i), i \in \{\text{img, txt}\}. \tag{1}$$

Without loss of generality, we ignore the subscript i in the following definitions unless otherwise specified. We split the lexical encoder in three stages: 1) The single-modal pre-trained feature extractor f, takes as input $\boldsymbol{x}$ and as output a feature sequence $\boldsymbol{y} := f(\boldsymbol{x}) \in \mathbb{R}^{n \times d_i}$, where n is the sequence length and d_i is the modal-dependent feature dimension; 2) The projector g, projects $\boldsymbol{y}$ to the lexical feature sequence $\boldsymbol{z} := g(\boldsymbol{y}) \in \mathbb{R}^{n \times d}$, where d is the lexical feature dimension; 3) The mapper h, maps $\boldsymbol{z}$ to the lexical representation $\boldsymbol{s} = h(\boldsymbol{z}) \in \mathbb{R}^V$.

In this paper, we restrict the lexical representation to be non-negative and ℓ_2-normalized, i.e., $\|\boldsymbol{s}\|_2 = 1, s_i \geq 0$. Different modalities share the same $\boldsymbol{V}$ with a unique codebook $\boldsymbol{Z}_i \in \mathbb{R}^{V \times d}, i \in \{\text{img, txt}\}$.

3.2.1 Lexical text encoder

Captioning? In VLA training data pair, the text is typically a caption of the corresponding image. However, it is well-known that caption only captures a partial observation of the semantics in the image, and cannot serve as a perfect target to represent the image. As it is the only accessible ground-truth signal, previous approaches mainly train the text encoders to represent the caption embedding via the output hidden state [9, 48, 12, 25] or [CLS] token embeddings [34], and use it as the target for the corresponding image. Hence, the learned alignment is biased and leads to false-elimination of image patterns that are not observed in the captions.

Predicting! Inspired by the powerful Large Language Models (LLMs), we would like to investigate *can we unleash the inherent knowledge of LLMs for lexical representation?* The answer is yes.

As the auto-regressive LLMs [1, 40] are not naturally to summarize the previous tokens, it is irrational to directly use the predicted token as the embedding for raw text inputs. A straightforward way to activate LLM is to use prompt like "summarize the sentence of [TEXT] in one word:" to activate LLM's summarization capacity and use the output token as the text embedding. But it is still the caption-style summarization, and cannot fully utilize the LLM.

LLM as lexical predictor Given the in-context learning capacity of LLM, we propose to adapt LLM as a lexical predictor. Specifically, our input for the LLM is:

> The focus of "The man is riding a white horse." lies on important words:"man", "riding", "white", "horse". The focus of "`[TEXT]`" lies on important words:

The `[TEXT]` is the input caption. Then we adopt the output token as the text embedding. This prompt includes two parts: 1) One in-context example to guide the LLM to identify the important words in the caption, adapting the LLM to lexical prediction task; 2) The question prompt to ask the LLM the important words of input caption. In this design, the output token of the LLM naturally performs the lexical prediction task to calculate the similarity between the input caption and every word in the vocabulary, and thus serves as a powerful proxy to get the lexical representation.

Realization we utilize Llama 2 [40] as our text encoder f_{txt}. The input text follows the above in-context prompts. We use the output of the predicted token embedding as $\boldsymbol{y}_{txt}$. As our lexical codebook is Llama 2's output codebook, we do not need g_{txt}, thus $\boldsymbol{z}_{txt} = \boldsymbol{y}_{txt}$. To implement h_{txt}, we first calculate dot-product attention between each text token $\boldsymbol{z}_{txt} \in \mathbb{R}^{1\times d}$ and the text lexical codebook $\boldsymbol{Z}_{txt}$. Then we use the $elu1p$ activation [48] to transform the attention score into non-negative values and then normalize it, yielding global lexical representation $\boldsymbol{s}_{txt}$. Formally,

$$\boldsymbol{s}_{txt} = \text{Normalize} \circ elu1p(\boldsymbol{z}_{txt}\boldsymbol{Z}_{txt}^{\top}), \tag{2}$$

where $elu1p(x) = (x+1)\cdot 1_{\{x\geq 0\}} + e^{x}\cdot 1_{\{x<0\}}$ and Normalize is the ℓ_2 normalization operator. For word-level lexical representation, we directly use the word code in $\boldsymbol{Z}_{txt}$ followed by Eq. (2).

3.2.2 Lexical visual encoder

We adopt DINOv2 [30] as our visual backbone to implement f_{img}. Given input $\boldsymbol{x}_{img}$, we first flatten it into patches $\boldsymbol{x}_{img} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}]$ and input to f_{img} to get

$$\boldsymbol{y}_{img} = f_{img}([\texttt{CLS}; \boldsymbol{x}_{\boldsymbol{img}}]) \in \mathbb{R}^{n\times d}, \tag{3}$$

where `CLS` is the "class" token in DINOv2. Subsequently, we implement g_{img} with an adapter includes a self-attention layer and 2 multi-layer perceptions to post-process the DINOv2 features. To implement h_{img}, similar to the text encoder but in a more fine-grained manner, we first calculate dot-product attention between each image patch token $\boldsymbol{z}_{img,i} \in \mathbb{R}^{1\times d}$ and the image lexical codebook $\boldsymbol{Z}_{img}$, followed by the $elu1p$ activation. In the end, we use max-pooling to aggregate patch representations and then normalize, yielding global lexical representation $\boldsymbol{s}_{img}$. Formally,

$$\boldsymbol{s}_{img} = \text{Normalize} \circ \text{Max-Pool} \circ elu1p(\boldsymbol{z}_{img}\boldsymbol{Z}_{img}^{\top}). \tag{4}$$

The image patch lexical representation follows the same process but replace the max-pooling operator with selecting the corresponding patch location.

https://doi.org/10.52202/079017-0035

3.3 Train LexVLA

Contrastive objective We use the InfoNCE loss [29] with learnable temperature τ for cross-modal retrieval over a batch of N text-image pairs (x_{img}, x_{txt}) as the major objective:

$$\ell_{a2b} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(a_i b_i^\top/\tau)}{\sum_{j=1}^{N}\exp(a_i b_j^\top/\tau)}, (a,b) \in \{(x_{img}, x_{txt}), (x_{txt}, x_{img})\}, \tag{5}$$

Overuse penalty In lexical representation learning, the FLOPs loss [31] is widely used to encourage the output sparsity. Given the modality-specified lexical representation matrix $\boldsymbol{S} = \{s_{i,j}\}, i \in \{1, \ldots, N\}, j \in \{1, \ldots, V\}$ in current mini-batch, the FLOPs loss is defined as:

$$\ell_{\text{FLOPs}} = \sum_{j=1}^{V}\bar{s}_{\cdot,j}^2 = \sum_{j=1}^{V}\left(\frac{1}{N}\sum_{i=1}^{N}s_{ij}\right)^2. \tag{6}$$

The FLOPs loss aims to reduce FLOPs to induce sparsity. But we noticed that sometimes this penalty pushes the encoder to take shortcuts that falsely activate rarely used and image-unrelated tokens. This semantic deviation leads to false activation and affects the interpretability of lexical representation.

To prevent using meaningless tokens too much, we adapt the FLOPs loss with a weighting method. Essentially, we aim to penalize tokens that are used too often across examples. We measure this via the normalized average activation value across the vocabulary, and propose the overuse penalty as:

$$\ell_{\text{overuse}} = V\sum_{j=1}^{V}\frac{\bar{s}_{\cdot,j}}{\sum_{k=1}^{V}\bar{s}_{\cdot,k}}\bar{s}_{\cdot,j}^2 = NV\sum_{j=1}^{V}\left(\sum_{i=1}^{N}s_{i,j}/N\right)^3 / \sum_{j=1}^{V}\sum_{i=1}^{N}s_{i,j}. \tag{7}$$

Objective The full objective contains Eq. (5) and Eq. (7) for each modality:

$$\mathcal{L} = \ell_{t2i} + \ell_{i2t} + \lambda_I \ell_{overuse}^I + \lambda_T \ell_{overuse}^T \tag{8}$$

where λ_I and λ_T are two regularization weights for image and text representations, respectively.

Incremental fine-tuning For text part, we adopt LoRA adapters [14] to fine-tune the Llama 2. For vision part, we freeze the DINOv2, and only train the projector and vision codebook.

3.4 PatchDis: interpretability metric

The core target of lexical representation is to achieve an interpretable feature representation for input modality. For the text input, the interpretability is straightforward to check. However, existing literature do not have a quantitative metric to measure the patch-level interpretability of visual lexical representation. In this paper, we propose the PatchDis metric to evaluate the patch-level discrimination tasks, inspired by the patch-level segmentation task but designed for models which are not trained from fine-grained alignment tasks like segmentation or detection.

Basically, PatchDis evaluates the classification task in the patch-level features. For any given VLA model, we use its text encoder to input the class names and obtain the text embeddings of all classes. Similarly, we input the testing image to the visual encoder to obtain all patch features. Then we could use the model-defined metric to calculate the similarity between each patch feature and the class embeddings, and predict the class of each patch from the largest similarity. Then for each class, we could aggregate all patches of one class as the patch-level prediction for the class. We use the ground-truth segmentation of the class as the target, and use mIoU as the metric to evaluate the patch-level interpretability of the VLA models in the zero-shot patch-level discrimination task.

4 Experiments

Datasets We use CC-12M [4] for training, a dataset consisting of 12.4 million image-text pairs. We successfully download 9.2M pairs and use this subset as our training set. For evaluation, we use Flickr30k [33] and MSCOCO [23] to evaluate zero-shot cross-modal retrieval tasks.

Table 1: Zero-shot cross-modal retrieval. 🔍 indicates variants of our LexVLA. CLIP[1] is the original CLIP [34]; results denoted by $(\cdot)^2$ are reported in VDR [48]; results denoted by $(\cdot)^3$ are reported in STAIR [5]. "Data" is the multi-modal alignment training data size; "Latent" means direct latent feature alignment methods; "Lexical" indicates lexical feature alignment methods. R@K, the recall ratio within top-K items.

Setting	Model	Data	MSCOCO						Flickr30k					
			image-to-text			text-to-image			image-to-text			text-to-image		
			R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Latent	CLIP[2]	15M	20.8	43.9	55.7	13.0	31.7	42.7	34.9	63.9	75.9	23.4	47.2	58.9
	FILIP[2]	15M	21.6	46.7	59.0	13.7	31.7	41.6	46.3	74.4	83.2	30.7	58.2	68.6
	CLIP-BERT[2]	15M	23.9	47.8	60.3	13.6	33.8	45.1	44.1	71.2	80.7	27.8	54.7	65.9
	DeCLIP[2]	15M	25.3	51.2	63.4	16.6	35.2	45.4	51.3	80.7	88.5	35.5	63.0	73.0
	SLIP[2]	15M	27.7	52.6	63.9	18.2	39.2	51.0	47.8	76.5	85.9	32.3	58.7	68.8
	ProtoCLIP[2]	15M	30.2	55.1	66.5	16.9	37.9	49.4	-	-	-	-	-	-
	CLIP[1]	0.4B	52.4	76.7	84.6	33.1	58.4	69.0	81.8	96.2	98.8	62.1	85.6	91.8
	CLIP[3]	1.1B	53.4	78.3	85.6	36.2	62.2	72.2	79.6	95.5	98.1	63.0	86.7	92.5
Lexical	VDR[2]	15M	30.9	54.5	65.4	17.4	38.1	49.7	51.0	79.3	86.7	32.4	60.1	70.7
	STAIR[3]	1.1B	**57.7**	80.5	87.3	**41.4**	65.4	75.0	81.2	96.1	98.4	66.6	88.7	93.5
Lexical	🔍(BoW)	12M	17.9	34.9	45.2	10.4	24.3	33.1	30.6	56.2	66.3	17.7	36.4	44.9
	🔍(CLIP)	12M	51.8	75.5	84.1	36.8	62.5	72.7	82.9	96.2	98.7	65.2	88.3	93.2
	🔍(FLOPs)	12M	56.2	80.0	87.4	39.0	65.7	75.6	84.2	96.6	98.7	67.4	89.4	94.1
	🔍(512)	12M	56.4	79.9	87.5	38.1	64.6	74.9	**84.5**	97.3	99.0	65.7	89.3	93.8
	LexVLA	12M	55.4	**80.6**	**88.3**	39.8	**66.3**	**76.2**	83.9	**97.5**	**99.1**	**67.8**	**90.2**	**94.2**

Implementation We use DINOv2 [30] base model and Llama 2 [40] 7B model as our backbones. We use Adam optimizer [17] with learning rate $5e-4$ and cosine decayed, batch size of 6,144, precision of BFloat16 for 12 epochs. We initialize τ as 0.07, and clip the logits larger than 100 as in [34]. We use 8 A100 GPUs of 40GB memory to train LexVLA. We quadratically warmup λ in the fisrt 2k steps and then freeze as [31]. We set λ_I as $5e-4$ and λ_T as $1e-3$. The trainable parameters in LexVLA is 109 M in total, including 70M for vision codebook, 17M for vision projector (19.76% compared with DINOv2), 21M for Llama adaptor (0.30% compared with Llama 2).

4.1 Zero-shot cross-modal retrieval

Evaluation We conduct experiments on zero-shot cross-modal retrieval tasks on Flickr30k and MSCOCO based on the splits in [16] following previous approaches. We use the R@K, recall ratio within top-K items, as the evaluation metric.

Competitors We compare LexVLA with the latent feature alignment methods, including the original CLIP [34] model trained on different dataset scales and followup CLIP-style alignment models, including FILIP [43], CLIP-BERT [48], DeCLIP [22], SLIP [26], and ProtoCLIP [6]. We also compare with previous lexical representation learning models that focus on zero-shot cross-modal retrieval tasks, including VDR [48] and STAIR [5]. Note that all the competitors are trained by larger datasets. Specifically, the models trained on 15M data uses YFCC15M [21], and models trained on 1.1B data [5] contains several datasets including CC-12M.

Variants For LexVLA, we report the following variants in addition to our final model: 1) 🔍(BoW), which selects only the nouns, adjectives, and non-auxiliary verbs in a caption (Bag-of-Words) instead of the LLM-based lexical predictor. 2) 🔍(CLIP), which uses the original CLIP visual model, instead of the proposed DINOv2 based encoder; 3) 🔍(FLOPs), which uses FLOPs loss as sparsity penalty. 3) 🔍(d), which uses the top-k thresholding strategy. We test $d = 512$, the same activated dimension as CLIP to make a fair comparison. The average activated dimension of the final LexVLA is 1081. Note that LexVLA surpasses CLIP with less activated values, see Fig. 5 for detailed analysis.

https://doi.org/10.52202/079017-0035

Figure 3: PatchDis visualization. The same color indicates the same category. LexVLA correctly predicts the corresponding region, even for the small-scale objects, like the bottle in the first image.

Figure 4: Visualization of the image lexical representation obtained by LexVLA. Larger word indicates larger lexical value. The first row represents the complete image, and the second row represent local patches (boxed in red). LexVLA learns a well-aligned lexical representation for both image and patches without local supervision.

Results We report the results in Table 1. **1)** LexVLA performs significantly better than both latent feature alignment methods and previous lexical alignment methods when trained on a similar amount of multi-modal examples (12M vs. 15M). This shows the effectiveness of LexVLA. **2)** Compared to CLIP trained on much larger multi-modal datasets (0.4B and 1.1B), LexVLA still performs better, even when using the same activated feature dimension as CLIP. This further demonstrates LexVLA's efficiency in text-image alignment with much less training data. **3)** LexVLA outperforms STAIR in most metrics and is comparable in two others, despite using significantly less training data (12M vs. 1.1B). This highlights LexVLA's training efficiency. **4)** The results of 🔍(BoW) are poor, due to the BoW method limits the model's ability to use information effectively. It struggles with vocabulary [7] and semantic [32] mismatches. LexVLA aims to address these limitations by leveraging language models for lexical prediction rather than relying solely on captioning. **5)** The significant improvements of LexVLA over 🔍(CLIP) indicate that using DINOv2 as the visual backbone is superior. **6)** LexVLA with FLOPs loss performs worse than our proposed overuse penalty, and activates meaningless tokens as in Fig. 6. In summary, these results demonstrate the effectiveness and efficiency of our method.

4.2 Lexical representation analysis

In this part, we evaluate if LexVLA learns the accurate lexical representation. We conduct the following experiments: 1) Quantitatively, we use the proposed PatchDis metric to evaluate the fine-grained patch-level visual lexical representations on MSCOCO 2017 [23]; 2) Qualitatively, we visualize the global lexical representation of images, local lexical representation of image patches, and a PatchDis visualization for the activated patches for the given category.

Table 2: PatchDis results.

Model	mIoU
Random Dis.	5.0
CLIP	5.3
VDR	12.6
🔍(CLIP)	13.9
LexVLA	**36.3**

PatchDis evaluation We report the zero-shot PatchDis mIoU performance on MSCOCO 2017 validation dataset. We compare LexVLA with public available baselines CLIP (0.4B) and VDR (15M), as well as random discrimination, and report the performance in Tab. 2. LexVLA clearly shows the superiority on the patch-level interpretability with the weak global training signal. In contrast, the latent alignment CLIP, due to no supervision on patch-level features, performs slightly better than random guessing. VDR enjoys better interpretability, but significantly worse than our LexVLA. LexVLA variant, using the original CLIP visual model, provides a better and more interpretable

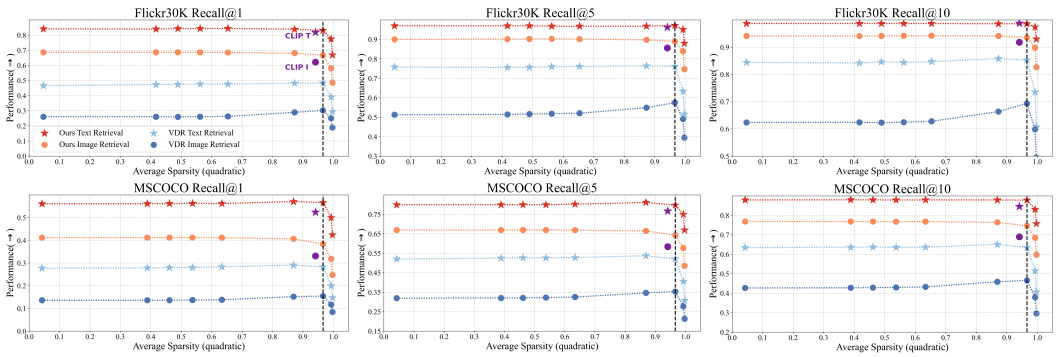

Figure 5: Retrieval in different sparse levels. We compare LexVLA with VDR in the same sparse levels and with CLIP as a proxy of dense latent alignment. The first row is results in the Flickr30K dataset, and the second in the MSCOCO dataset. The first to the third columns show the settings of Recall@1, Recall@5, and Recall@10, respectively. Purple symbols represent CLIP.

lexical representation of visual patches compared to both the original CLIP and previous lexical methods. This further confirms the effectiveness of our proposed approach. We also visualize the predicted patches of different classes in Fig. 3. LexVLA successfully activates the correlated patches in the image for different classes, even for the small-scale objects.

Lexical visualization We visualize the lexical representations of the global image and local patches in Fig. 4. For the bottom row, we randomly select one object in the image and then annotate the patches that cover the object to calculate the lexical representation on patches in the free form. We use word cloud [28] to illustrate the lexical representation where larger word indicates larger lexical value. These results clearly suggest that the learned lexical representation correctly correlates with the semantics of the image/patch, showcasing the effectiveness of LexVLA.

4.3 Further analysis

LexVLA under different sparsity One of the targets for lexical representation is to learn a sparse representation for potential benefits in large scale retrieval tasks. To test the robustness of learned representations in LexVLA, we test the retrieval performance of LexVLA in different sparsity levels. Particularly, the sparsity ratio is calculated via $\ell_0(s_i)/V$, the ratio of non-zero elements in the lexical representation s_i. We compare LexVLA with VDR on the same sparsity ratios. We also visualize the CLIP's performance as a proxy of dense latent representation.

Results are shown in Fig. 5. **1)** LexVLA is robust against the sparsity ratio even in a very high ratio (98.27%, 296 activated tokens, which is marked with black vertical dotted line in Fig. 5), and then starts to get damaged as the sparse ratio approaches 1; **2)** Compared with VDR, LexVLA achieves better performance in all sparse level, indicating a consistent improvement. **3)** LexVLA enjoys superior performance with less activated feature dimension as CLIP (296 vs 512) in almost all cases, showcasing our model's effectiveness.

Overuse penalty The widely used FLOPs loss aims to encourage the sparsity of learned lexical representation. However, we empirically find that it will falsely activate meaningless tokens as in Fig. 6. Specifically, the model tends to activate certain semantically unrelated tokens across different s_i (like '@' in the image and 'contemporary' in the text, respectively), implicitly treating these tokens as latent dimensions. Our proposed overuse penalty inhibits these frequently activated words more aggressively, and leads to more semantic-correlated lexical representations in Fig. 6. This enhanced interpretability further improves the retrieval performance in most cases, as in Tab. 1.

Limitations The major limitation of LexVLA is the use of vocabulary from large language model. We design our lexical vocabulary based on the Llama 2's tokenizer, and sometimes it splits a word into several sub-word tokens, making the resulting token not a complete word. Although we filter out many meaningless tokens, it still introduces the gap between our vocabulary and the perfect word-level vocabulary. Given that random initialize a word-level vocabulary and additionally learn a

https://doi.org/10.52202/079017-0035

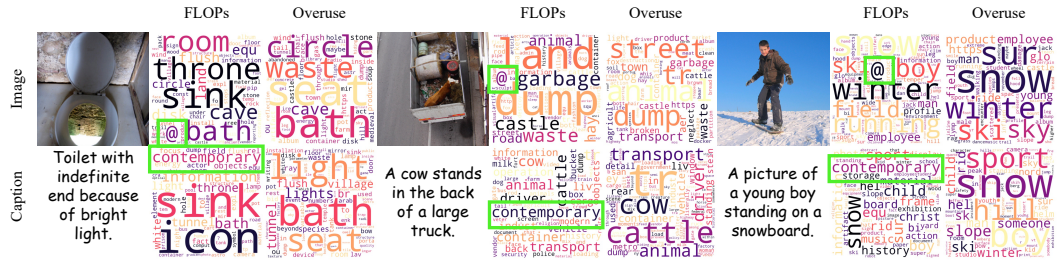

Figure 6: Visualization of lexical representations of LexVLA w/ FLOPs loss and w/ our proposed overuse penalty, respectively. The first row represents images while the second is caption. LexVLA w/ FLOPs loss usually falsely activates similar meaningless tokens frequently as a shortcut (see '@' and 'contemporary' outlined by green boxes as examples). LexVLA w/ overuse penalty significantly mitigate this false activation issue.

projector from sub-word tokens to word-level tokens works poorly, we regard designing a word-level vocabulary that still benefit from the LLMs as a future work.

Broader impacts The advancements made in cross-modal retrieval, as demonstrated in this work, hold significant potential for various fields and applications. One immediate impact lies in enhancing information retrieval systems, where users can efficiently search and retrieve multimodal content such as images and text. This can greatly benefit industries such as e-commerce, digital libraries, and multimedia databases, enabling more intuitive and accurate searches. While the advancements in cross-modal retrieval offer numerous benefits, one concern is related to privacy and security, as the integration of multimodal data may raise issues regarding data protection and user consent. It's essential to mitigate these potential negative impacts through robust privacy measures.

5 Conclusion

In this paper, we propose LexVLA, a unified lexical vision-language alignment framework. LexVLA effectively learns a unified lexical representation with unique codebooks for vision and language modalities, with the aid of single-modal pre-trained models. We propose to incremental fine-tune LexVLA with the standard contrastive objective penalized by the proposed overuse objective to prevent meaningless activation and encourage sparse embedding. LexVLA refrains from the complex training configurations, and effectively achieves the patch-level interpretability with only global supervision signals. We introduce a patch-level interpretability metric, PatchDis, to quantify this on the zero-shot cross-modal benchmark dataset. We demonstrate the effectiveness of LexVLA on two zero-shot cross-modal retrieval benchmark datasets with significantly smaller multi-modal training dataset. We also provide detailed analysis on LexVLA.

References

[1] 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.

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

[3] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). *arXiv preprint arXiv:2402.10376*, 2024.

[4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.

[5] Chen Chen, Bowen Zhang, Liangliang Cao, Jiguang Shen, Tom Gunter, Albin Madappally Jose, Alexander Toshev, Jonathon Shlens, Ruoming Pang, and Yinfei Yang. Stair: Learning sparse text and image representation in grounded tokens. *arXiv preprint arXiv:2301.13081*, 2023.

[6] Delong Chen, Zhao Wu, Fan Liu, Zaiquan Yang, Yixiang Huang, Yiping Bao, and Erjin Zhou. Prototypical contrastive language image pretraining. *arXiv preprint arXiv:2206.10996*, 2022.

[7] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading, 2010.

[8] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[10] Sivan Doveh, Assaf Arbelle, Sivan Harary, Roei Herzig, Donghyun Kim, Paola Cascante-Bonilla, Amit Alfassy, Rameswar Panda, Raja Giryes, Rogerio Feris, et al. Dense and aligned captions (dac) promote compositional reasoning in vl models. *Advances in Neural Information Processing Systems*, 36, 2024.

[11] Jiali Duan, Liqun Chen, Son Tran, Jinyu Yang, Yi Xu, Belinda Zeng, and Trishul Chilimbi. Multi-modal alignment using representation codebook. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15651–15660, 2022.

[12] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292, 2021.

[13] Luyu Gao, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv preprint arXiv:2104.07186*, 2021.

[14] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[15] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's" up" with vision-language models? investigating their struggle with spatial reasoning. *arXiv preprint arXiv:2310.19785*, 2023.

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

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[19] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, 2018.

[20] 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*, pages 19730–19742. PMLR, 2023.

 https://doi.org/10.52202/079017-0035

[21] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021.

[22] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International Conference on Learning Representations*, 2022.

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

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

[25] Ziyang Luo, Pu Zhao, Can Xu, Xiubo Geng, Tao Shen, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Lexlip: Lexicon-bottlenecked language-image pre-training for large-scale image-text sparse retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11206–11217, 2023.

[26] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European conference on computer vision*, pages 529–544. Springer, 2022.

[27] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[28] Layla Oesper, Daniele Merico, Ruth Isserlin, and Gary D Bader. Wordcloud: a cytoscape plugin to create a visual semantic summary of networks. *Source code for biology and medicine*, 6(1):7, 2011.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[30] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[31] Biswajit Paria, Chih-Kuan Yeh, Ian E.H. Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. Minimizing flops to learn efficient sparse representations. In *International Conference on Learning Representations*, 2020.

[32] ME Peters, M Neumann, M Iyyer, M Gardner, C Clark, K Lee, and L Zettlemoyer. Deep contextualized word representations. corr. *arXiv preprint arXiv:1802.05365*, 2018.

[33] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

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

[35] 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*, pages 10684–10695, 2022.

[36] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, 2016.

[37] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

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

[39] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024.

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

[41] Fei Wang, Liang Ding, Jun Rao, Ye Liu, Li Shen, and Changxing Ding. Can linguistic knowledge improve multimodal alignment in vision-language pretraining? *arXiv preprint arXiv:2308.12898*, 2023.

[42] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023.

[43] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2022.

[44] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[45] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022.

[46] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 497–506, 2018.

[47] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022.

[48] Jiawei Zhou, Xiaoguang Li, Lifeng Shang, Xin Jiang, Qun Liu, and Lei Chen. Retrieval-based disentangled representation learning with natural language supervision. In *The Twelfth International Conference on Learning Representations*, 2024.

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

 https://doi.org/10.52202/079017-0035

A Appendix / supplemental material

A.1 Model hyperparameters

Our hyperparamters selection for the detail can be found in Tab. 3

Table 3: Hyperparameters for training LexVLA.

Hyperparameter	**Value**
Batch size	6144
Learning rate	5e-4
Vocabulary size	17149
Training epochs	12
Max temperature τ	100.0
Warm-up iterations	1000
LR Scheduler	Cosine
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	1e-6
Penalty $lambda_1$	5e-4
Penalty λ_2	1e-3
Lora_alpha	16
Lora_r	8
Lora_dropout	0.05

A.2 Pretraining details of backbone models

The DINOv2 [30] base model used in LexVLA is pretrained with self-supervised learning, leveraging a large dataset of 142 million diverse, high-resolution images. It employs a teacher-student architecture with multi-crop views and strong data augmentations to encourage robustness and capture both global and local features. During pretraining, the model uses a cosine similarity loss between student and teacher outputs to ensure consistent feature representations across different augmentations. This model supports downstream performance on detection and segmentation tasks.

The Llama 2 model [40] used in LexVLA is pretrained with self-supervised learning on approximately 2 trillion tokens from a diverse mix of publicly available datasets and web-sourced content. It employs a dense transformer architecture, and during training, it focuses on token prediction accuracy and generalization. This model supports effective performance across various language tasks.

For more pretraining details, please refer to the original papers of these two pretrained models.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, they correctly describe the paper's contributions and scope. We propose a unified lexical alignment model that aligns text and visual data based on single-modal pre-trained models and introduces unique codebooks, an overuse penalty, and enjoys patch-level interpretability.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, we discussed the limitations in Sec. 4.3.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-0035

Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we provided all the information needed to reproduce the main experimental results in Sec. 4 and Sec. A.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: (1) We used publicly available benchmark datasets and have provided references for each one for open access. (2) According to the rules of the authors' organization, the code will be available only after the official publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provided all the details in Sec. 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We didn't include error bars because calculating them would take too much computing power. We're also following the approach of previous studies that didn't include them.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/079017-0035

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided all the required information in Sec. 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed both potential positive societal impacts and negative societal impacts in Sec. 4.3.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work, towards the accurate lexical alignment between text and images, poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have credited all the assets we used, followed their licensing rules, and will not distribute or repurpose them.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

https://doi.org/10.52202/079017-0035

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not have new assets in the submission.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-0035