REASONING, SCALING, GENERATING WITH VISION-LANGUAGE MODELS

BY

ZHONGHAO WANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2024

Urbana, Illinois

Doctoral Committee:

Professor Mark Hasegawa-Johnson, Chair
Associate Professor Humphrey Shi, Chair, Georgia Tech
Associate Professor Lav Varshney
Dr. Wei Wei, Accenture

ABSTRACT

The intersection of vision and language models has paved the way for groundbreaking advancements in artificial intelligence, enabling systems to comprehend and generate multimodal content with unprecedented sophistication. This dissertation presents a comprehensive study on enhancing the capabilities of vision-language models (VLMs) with a focus on three critical dimensions: reasoning, scaling, and generating. Through an innovative amalgamation of deep learning techniques, this research advances the understanding and application of VLMs in performing complex reasoning tasks, scaling to accommodate diverse and large-scale datasets, and generating coherent and contextually relevant multimodal outputs.

Firstly, the dissertation introduces a novel framework for augmenting VLMs with enhanced reasoning capabilities, allowing them to infer and deduce information from visual and textual cues in a manner akin to human cognitive processes. Specifically, we study the problem of concept induction in visual reasoning, i.e., identifying concepts and their hierarchical relationships from question-answer pairs associated with images; and we achieve an interpretable model via working on the induced symbolic concept space. To this end, we first design a new framework named object-centric compositional attention model (OCCAM) to perform the visual reasoning task with object-level visual features. Then, we come up with a method to induce concepts of objects and relations using clues from the attention patterns between objects' visual features and question words. Finally, we achieve a higher level of interpretability by imposing OCCAM on the objects represented in the induced symbolic concept space. Experiments on the CLEVR and GQA datasets demonstrate: 1) our OCCAM achieves a new state of the art without human-annotated functional programs; 2) our induced concepts are both accurate and sufficient as OCCAM achieves an on-par performance on objects represented either in visual features or in the induced symbolic concept space.

Secondly, the dissertation addresses the challenge of scaling VLMs, both in terms of model architecture and data handling. We propose a multi-task model architecture that improve performances for multiple downstream video tasks including temporal action localization, moment retrieval, and action segmentation. While large-scale image-text pretrained models such as CLIP have been used for multiple video-level tasks on trimmed videos, their use for temporal localization in untrimmed videos is still a relatively unexplored task. We design a new approach for this called UnLoc, which uses pretrained image and text towers, and feeds tokens to a video-text fusion model. The output of the fusion module are then used to construct a feature pyramid in which each level connects to a head to predict a per-frame relevancy score and start/end time displacements. Unlike previous works, our architecture enables Moment Retrieval, Temporal Localization, and Action Segmentation with a single stage model, without the need for action proposals, motion based pretrained features or representation masking. Unlike specialized models, we achieve state of the art results on all three different localization tasks with a unified approach.

Lastly, the dissertation delves into the generation capabilities of VLMs, presenting methodologies for creating accurate and diverse visual content in accordance with textual descriptions. We explore advancements in high-fidelity personalized image generation through the utilization of pre-trained text-to-image diffusion models. While previous approaches have made significant strides in generating versatile scenes based on text descriptions and a few input images, challenges persist in maintaining the subject fidelity within the generated images. In this work, we introduce an innovative algorithm named HiFi Tuner to enhance the appearance preservation of objects during personalized image generation. Our proposed method employs a parameter-efficient fine-tuning framework, comprising a denoising process and a pivotal inversion process. Key enhancements include the utilization of mask guidance, a novel parameter regularization technique, and the incorporation of step-wise subject representations to elevate the sample fidelity. Additionally, we propose a reference-guided generation approach that leverages the pivotal inversion of a reference image to mitigate unwanted subject variations and artifacts. We further extend our method to a novel image editing task: substituting the subject in an image through textual manipulations. Experimental evaluations conducted on the DreamBooth dataset using the Stable Diffusion model showcase promising results. Fine-tuning solely on textual embeddings improves CLIP-T score by 3.6 points and improves DINO score by 9.6 points over

Textual Inversion. When fine-tuning all parameters, HiFi Tuner improves CLIP-T score by 1.2 points and improves DINO score by 1.2 points over DreamBooth, establishing a new state of the art.

This dissertation represents a significant step forward in the quest to build more intelligent vision-language models, offering insights and tools that will fuel future innovations in the field.

To my parents, for their love and support.

ACKNOWLEDGMENTS

Completing this dissertation has been a monumental journey, and it would not have been possible without the support and encouragement of many people. First and foremost, I extend my deepest gratitude to my supervisors, Mark Hasegawa-Johnson, Humphrey Shi and Thomas Huang, whose expertise, understanding, and patience, added considerably to my graduate experience. They not only provided academic guidance but also encouraged me to challenge myself and explore uncharted territories in my research field. The time spent discussing concepts, reviewing drafts, and strategizing on research directions have been pivotal in shaping this work.

I am also profoundly thankful to my dissertation committee, comprised of Prof. Mark Hasegawa-Johnson, Prof. Humphrey Shi, Prof. Lav Varshney and Dr. Wei Wei. Each member brought a unique perspective and expertise that enriched my understanding and approach to my research topic. Their rigorous feedback pushed me to refine my arguments and deepen my analysis.

My sincere thanks also go to members of Image Formation Group (IFP). I remember spending days and nights doing experiments with them in the lab. They provided tremendous emotional support to my research.

I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this dissertation. This accomplishment would not have been possible without them. I want to thank my father, Renkun Wang, and my mother, Wei Lu, for believing in me.

This journey has been a test of perseverance, patience, and hard work, and I am grateful for the opportunity to learn and grow in this process.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This dissertation embarks on an in-depth exploration of the burgeoning field of Vision-Language Models (VLMs), focusing on three pivotal aspects: reasoning, scaling, and generating. In an era where artificial intelligence (AI) increasingly intersects with human life, the development of models that can understand and interpret the world in ways that mimic human cognitive processes has become paramount. VLMs, standing at the crossroads of computer vision and natural language processing, represent a significant leap forward in this endeavor, offering unprecedented opportunities to bridge the semantic gap between visual information and linguistic expression.

The introduction of VLMs has opened new frontiers in AI, enabling applications ranging from automated content creation to sophisticated interactive systems that can engage in meaningful dialogues with users. However, as with any rapidly advancing technology, the journey is fraught with challenges. This dissertation, therefore, is dedicated to dissecting these challenges through the lenses of reasoning, scaling, and generating—three areas critical to the advancement of VLMs.

First, we delve into the aspect of **reasoning**, examining how VLMs can be designed to not only perceive and understand visual and textual information but also to infer, deduce, and reason over these modalities in a manner that mirrors human thought processes. The ability to reason allows models to make sense of complex, ambiguous environments and interact with users in a more intelligent and contextualized way. We propose a method that induces a symbolic space alongside a conventional neural network, enabling better interpretability of the model's reasoning process.

We begin by highlighting the importance of interpretability in visual reasoning models. We argue that while deep learning models have achieved remarkable performance in various tasks, their decision-making processes often lack transparency. This opacity can be problematic, especially in critical applications such as healthcare or autonomous vehicles, where understanding the model's reasoning

is essential for trust and safety. Therefore, there is a growing need for interpretable visual reasoning models that can provide insights into their decision-making process.

To address this challenge, We propose a new framework that combines neural networks with a symbolic space. The symbolic space represents high-level reasoning concepts in a structured and interpretable manner, enabling humans to understand and interpret the model's decisions. Unlike traditional neural networks that operate on raw pixel values, the proposed method learns to map visual inputs into the induced symbolic space, where reasoning and decision-making occur in a more interpretable form.

The main idea behind the proposed method is to learn two components simultaneously: a neural network for feature extraction and a symbolic space for reasoning. The neural network takes raw visual inputs and extracts high-dimensional feature representations, while the symbolic space encodes high-level concepts such as object categories, relations, and spatial arrangements. By jointly optimizing these components, the model learns to perform visual reasoning tasks while maintaining interpretability through the induced symbolic space.

We evaluate our method on several benchmark visual reasoning datasets, including CLEVR and GQA. We compare our approach against baseline models and demonstrate superior performance in terms of both accuracy and interpretability. Overall, we present a promising approach to enhance the interpretability of visual reasoning models by inducing a symbolic space alongside conventional neural networks. By learning to reason in this structured space, the model can provide insights into its decision-making process, enabling humans to understand and trust its outputs. This work represents an important step towards building more transparent and interpretable artificial intelligence systems, with potential applications in various domains where visual reasoning is critical.

Next, we address the challenge of **scaling**. As VLMs grow in sophistication, so too do their demands for computational resources and data. Scaling these models in a manner that is both efficient and effective is crucial for their practical application and further development. We focus on the challenge of video localization - the ability to accurately identify and classify temporal segments within videos based on specific criteria or textual queries. Traditional approaches to this task have often relied on task-specific models, each tailored to address particular aspects of video localization such as Moment Retrieval, Temporal Action Localization, or Action Segmentation. While effective to a degree, these specialized models

often operate in silos, necessitating distinct architectures and training paradigms for each task. This fragmentation not only complicates the development and deployment of scalable video understanding systems but also limits the potential for cross-task knowledge transfer and efficiency gains.

We present a framework named UnLoc that reimagines the approach to video localization through the lens of a unified framework. Motivated by the limitations of existing methodologies and the untapped potential of large-scale image-text pretrained models like CLIP, this research endeavors to harness the rich semantic understanding encapsulated in these models to address the multifaceted challenge of video localization across different tasks within a single, cohesive architecture.

At the heart of UnLoc lies an innovative architecture that seamlessly integrates pretrained image and text towers with a video-text fusion model. This fusion model acts as the cornerstone of UnLoc, enabling the framework to process and synthesize video and textual information in a manner that is both contextually aware and temporally precise. The subsequent construction of a feature pyramid from the fusion module's output represents a novel strategy for capturing and classifying temporal segments at varying scales, facilitated by a hierarchical arrangement of prediction heads tasked with generating per-frame relevancy scores and temporal displacements.

This research makes several significant contributions to the field of video content analysis. Firstly, UnLoc introduces a unified model capable of performing Moment Retrieval, Temporal Action Localization, and Action Segmentation without the need for task-specific modifications or auxiliary mechanisms such as action proposals and representation masking. Secondly, the framework sets new benchmarks in performance across these tasks, demonstrating the efficacy of a unified approach over traditional, specialized models.

The implications of UnLoc extend far beyond the technical achievements detailed in this work. By providing a versatile and efficient framework for video localization, UnLoc paves the way for the development of more intelligent, scalable, and cohesive video understanding systems. Such systems have the potential to revolutionize a wide range of applications, from enhancing security and surveillance operations to enabling more dynamic and context-aware content discovery and management solutions.

Finally, the dissertation explores **generating** capabilities of VLMs. The power of VLMs to create coherent, contextually relevant and novel content—be it textual descriptions, visual imagery, or a combination thereof—has vast implications for

creative industries, educational tools, and beyond. In this work, we mainly focus on the text-to-image generation. Previous works using diffusion models enable the generation of highly realistic images from textual descriptions. Despite their success, these models often struggle with maintaining high fidelity to specific subjects when generating personalized images. We address this issue by proposing an innovative fine-tuning approach named HiFi Tuner that significantly enhances the subject fidelity of generated images.

The motivation behind the HiFi Tuner stems from the inherent limitations of existing text-to-image models. While capable of producing diverse and complex scenes, these models frequently fail to accurately preserve the appearance and characteristics of particular subjects across different generated images. This limitation hampers their applicability in scenarios where consistent subject representation is crucial, such as personalized content creation and digital art generation.

The core contribution of the HiFi Tuner is a novel algorithm designed to improve the preservation of object appearances during personalized image generation. This method leverages a parameter-efficient fine-tuning framework that integrates a denoising process and a pivotal inversion process. The innovation includes the application of mask guidance, a new parameter regularization technique, and the incorporation of step-wise subject representations to enhance the fidelity of generated samples. Another critical aspect of the HiFi Tuner is its reference-guided generation approach. This technique uses the pivotal inversion of a reference image to control and reduce unwanted variations and artifacts in the subject of the generated images. This approach is particularly beneficial for maintaining the consistency and accuracy of subject representation across different images.

The HiFi Tuner was rigorously evaluated using the DreamBooth dataset in conjunction with the Stable Diffusion model. The evaluation focused on comparing the performance of the HiFi Tuner against established benchmarks such as Textual Inversion and the original DreamBooth approach. The results were quantified using the CLIP-T score and DINO score, two metrics that measure the fidelity and coherence of the generated images in relation to the input descriptions and reference images. The experiments demonstrated that the HiFi Tuner significantly outperforms existing methods in terms of subject fidelity. Fine-tuning solely on textual embeddings resulted in notable improvements over Textual Inversion, with a 3.6 point increase in the CLIP-T score and a 9.6 point rise in the DINO score. When fine-tuning all parameters, the HiFi Tuner achieved even better results, sur-

passing DreamBooth and setting new state-of-the-art performance levels.

These findings underscore the potential of the HiFi Tuner to redefine the landscape of personalized image generation. By enhancing the accuracy and consistency of subject representation, the HiFi Tuner opens up new possibilities for applications in digital art, personalized content creation, and beyond.

This dissertation aims to contribute to the understanding of VLMs by addressing these three critical aspects: reasoning, scaling, and generating. Through a combination of theoretical analysis, model development, and evaluation, this work seeks to advance the state of the art in vision-language integration and open new pathways for future research and application.

CHAPTER 2

RELATED WORKS

2.1 Visual Question Answering (VQA)

Visual Question Answering (VQA) requires models to reason a question about an image to infer an answer. Recent VQA approaches can be partitioned into two groups: holistic models [1, 2, 3, 4, 5] and modular models [6, 7, 8, 9, 10, 11, 12], according to whether the approach has explicit sub-task structures. A typical holistic model, MAC [5], perform iterative reasoning steps with an attention mechanism on the image. A modular framework, NS-CL [12], designs multiple principle functions over the extracted features to explain the reasoning process.

2.2 Scene Graph Grounding

Scene graph grounding requires to construct the relationship among objects in an image. [13] designs a graph R-CNN model to detect objects and classify relations among them simultaneously. [14] uses graphs to ground words and phrases to image regions. [15] proposes to link words to image concepts in an unsupervised setting. However, all these works have predefined object and relation concepts. We focus on inducing the concepts from the language compositionality to better interpret the reasoning framework.

2.3 Model Interpretability

Model interpretability aims to explain the neural model predictions. [16] proposed network dissection to quantify interpretability of CNNs. [17] explains a CNN at the semantic level with decision trees. [18] generates scene graphs from images to explicitly trace the reasoning-flow. [10, 19] focused on visual attentions to pro-

vide enhanced interpretability. Our work is closely related to the self-explaining systems via rationalization [20, 21, 22]. These works usually extract subsets of inputs as explanations, while our work moves one-step further by learning parts of the structural explanation definitions (i.e., our concept hierarchy) together with explanations (i.e., the concept-level reasoning flow).

2.4 Visual Concept Learning

Visual concept learning contributes to broad visual-linguistic applications, such as cross-modal retrieval [23], visual captioning [24], and visual-question answering [25, 26]. [11, 12] attempt to disentangle visual concept learning and reasoning. Based on the visual concepts learned from VQA, [27] learns metaconcepts, i.e., relational concepts about concepts, with augmented QA-pairs about metaconcepts. Our work differs from the previous ones in learning concepts and super concepts without external knowledge.

2.5 Temporal Action Localization (TAL)

Supervised learning-based temporal action localization can be summarized into two-stage [28, 29, 30, 31, 32] and single-stage methods [33, 34, 35, 36]. More recently, EffPrompt [37] uses a two-stage sequential localization and classification architecture for zero-shot action localization, with the first stage consisting of action proposal generation with an off-the-shelf pre-trained proposal detector (e.g., BMN [31]), followed by proposal classification using CLIP features. We aim to build a proposal-free framework and directly regress the temporal location of the corresponding class labels or queries by using the fused video-text features. The closest to our method is STALE [38], which trains a single-stage model for zero-shot localization and classification, using representation masking for frame level localization. Unlike STALE, which evaluates on only TAL, we present a single unified method for MR, TAL and AS, and also introduce a feature pyramid for multi-scale reasoning.

2.6 Moment Retrieval (MR)

Unlike TAL, where class names are predefined used a closed-form vocabulary, MR aims to find the relevant clip in an untrimmed video for a given open-ended natural language query. Early works use sliding windows over video sequences to generate video segment proposals [39, 40], after which the proposals are ranked by their similarity to the query. This ignores the finegrained relationships between video frames and the words in sentences. Anchor-based methods [41, 42, 43] avoid proposal generation by assigning each frame with multi-scale anchors sequentially and use these to obtain more finegrained matchings between video and text. Regressionbased methods [44, 45, 46, 47, 48, 49] involve learning cross-modal interactions to directly predict the temporal boundaries of a target moment without the need for proposal generation. Our work belongs to this category; unlike works that tend to use the text tower only at the end to compute similarity scores [39, 50, 51, 46, 49], we fuse image and text tokens early on in our model to better leverage language priors from the pretrained CLIP text tower.

2.7 Action Segmentation (AS)

Action segmentation involves assigning a pre-defined label to each token or frame in a untrimmed long video, which helps to distinguish meaningful video segments from other tokens or frames [52]. While previous works [53, 54, 55, 56] pretrained their models on HowTo100M [57], our approach involves initializing models with pretrained CLIP models. CLIP was trained on pairs of web images and text, which may be less prone to noise compared to ASR and clip pairs.

2.8 Subject-driven text-to-image generation

This task requires the generative models generate the subject provided by users in accordance with the textual prompt description. Pioneer works [58, 59] utilize Generative Adversarial Networks (GAN) [60] to synthesize images of a particular instance. Later works benefit from the success of diffusion models [61, 62] to achieve a superior faithfulness in the personalized generation. Some works [63, 64] rely on retrieval-augmented architecture to generate rare subjects. How-

ever, they use weakly-supervised data which results in an unsatisfying faithfulness for the generated images. There are encoder-based methods [65, 66, 67] that encode the reference subjects as a guidance for the diffusion process. However, these methods consume a huge amount of time and resources to train the encoder and does not perform well for out-of-domain subjects. Other works [68, 69] finetune the components of diffusion models with the provided subject images. Our method follows this line of works as our models are faithful and generic in generating rare and unseen subjects.

2.9 Text-guided image editing

This task requires the model to edit an input image according to the modifications described by the text. Early works [70, 69] based on diffusion models [61, 62] prove the effectiveness of manipulating textual inputs for editing an image. Further works [71, 72] propose to blend noise with the input image for the generation process to maintain the layout of the input image. Prompt-to-Prompt [73, 74] manipulates the cross attention maps from the image latent to the textual embedding to edit an image and maintain its layout. InstructPix2Pix [75] distills the diffusion model with image editing pairs synthesized by Prompt-to-Prompt to implement the image editing based on instructions.

CHAPTER 3

INTERPRETABLE VISUAL REASONING VIA INDUCED SYMBOLIC SPACE

3.1 Introduction

Recent advances in Visual Question Answering (VQA) [3, 4, 5, 7, 8, 9, 10, 11, 12] usually rely on carefully designed neural attention models over images, and rely on pre-defined lists of concepts to enhance the compositional reasoning ability of the attention modules. Human prior knowledge plays an essential role in the success of the model design.

We focus on a less-studied problem in this field – given only question-answer pairs and images, induce the visual concepts that are sufficient for completing the visual reasoning tasks. By sufficiency, we hope to maintain the predictive accuracy for VQA when using the induced concepts in place of the original visual features. We consider concepts that are important for visual reasoning, including properties of objects (e.g., *red*, *cube*) and relations between objects (e.g., *left*, *front*). The aforementioned scope and sufficiency criterion require accurately associating the induced symbols of concepts to both visual features and words, so that each new instance of question-image pair can be transformed into the induced concept space for further computations. Additionally, it is necessary to identify super concepts, i.e., hypernyms of concept subsets (e.g., *shape*). The concepts inside a super concept are exclusive, so that the system knows each object can only possess one value in each subset. This introduces structural information to the concept space (multiple one-hot vectors for each visual object) and further guarantees the accuracy of the aforementioned transformation.

The value of the study has two folds. First, our proposed problem aims to identify visual concepts, their argument patterns (properties or relations) and their hierarchy (super concepts) *without* using any concept-level supervision. Solving the problem frees both the efforts of human annotations and human designs of concept schema required in previous visual reasoning works. At the same time, the

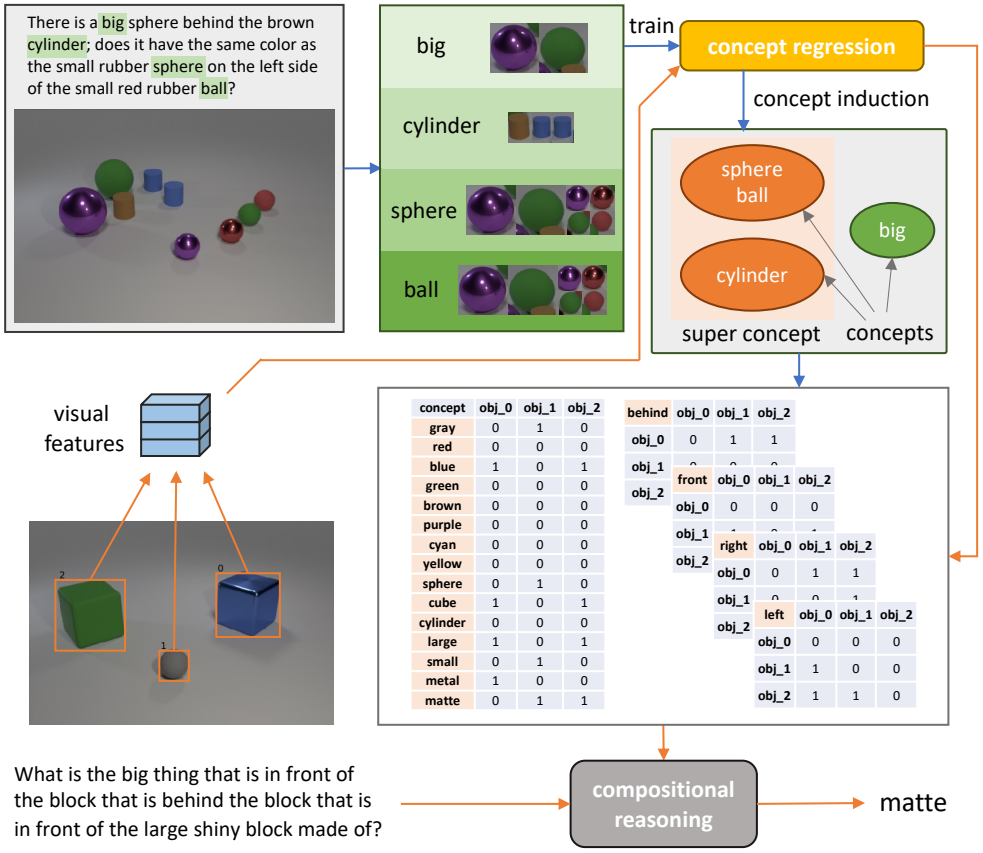


Figure 3.1: Illustration of our framework. Our model induces the concepts and super concepts with the attention correlation between the objects and question words in image-question pairs as the paths shown in blue arrows. Then, it answers a question about an image via compositional reasoning on the induced symbolic representations of objects and object relations, shown as the orange paths.

problem is technically more challenging compared to the related existing problem like unsupervised or weakly-supervised visual grounding [15]. Second, by constraining the visual reasoning models to work over the induced concepts, the ability of concept induction improves the interpretability of visual reasoning models. Unlike previous interpretable visual reasoning models that rely on human-written rules to associate neural modules with given concept definitions [10, 12, 18], our method resolves the concept definitions and associations interpretability automatically in the learning process, without the need of trading off for hand-crafted model designs.

We achieve the proposed task in three steps. First, we propose a new model architecture, object-centric compositional attention model (OCCAM), that performs object-level visual reasoning instead of pixel-level by extracting object-level vi-

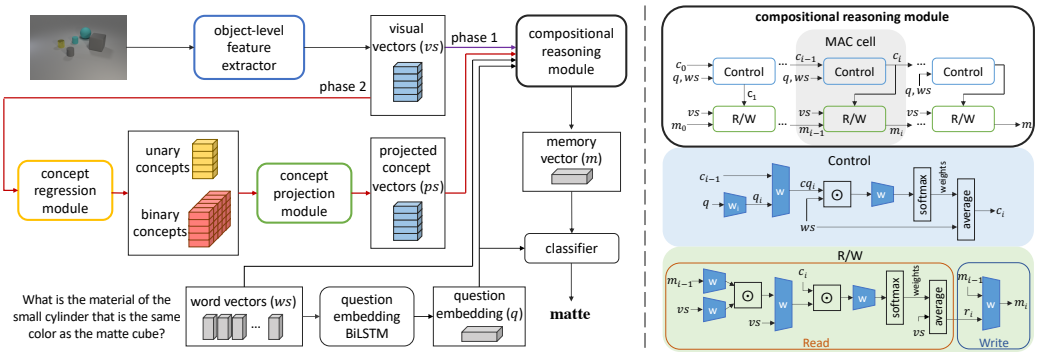


Figure 3.2: The framework and the compositional reasoning module. The left graph shows the general framework; The phase 1 training path is drawn in purple and the phase 2 training paths are drawn in red. The black paths are shared for both training phases. The structures of our proposed object-level feature extractor, concept regression module and concept projection module are shown in Figures 3.3, 3.4 and 3.7.

sual features with ResNet [76] and pooling the features according to each object's bounding box. The object-level reasoning not only improves over the state-of-the-arts, but also provides a higher-level interpretability for concept association and identification. Second, we benefit from the trained OCCAM's attention values over objects to create classifiers mapping visual objects to words and then derive the concepts and super concepts from the object-word cooccurrence matrices as shown in Figure 3.1. Finally, our concept-based visual reasoning framework predicts the concepts of objects and object relations, and then performs compositional reasoning using the predicted symbolic concept embeddings instead of the original visual features.

Experiments on the CLEVR and GQA datasets confirm that our overall approach improves the interpretability of neural visual reasoning, and maintains the predictive accuracy: (1) our OCCAM improves over the previous state-of-the-art models that do not use external training data; (2) our induced concepts and concept hierarchy are accurate in human study; and (3) our induced concepts are sufficient for visual reasoning – replacing visual features with concepts leads to only $\sim$1% performance drop.

3.2 OCCAM: Object-Centric Visual Reasoning

This section introduces a new neural architecture, object-centric compositional attention model (OCCAM), that performs visual reasoning over the object-level

visual features. This model not only achieves state-of-the-art performance, but also plays a key role in inducing object-wise or relational concepts as will be described in section 3.3.

Figure 3.2 shows our general framework with two training phases, each consists to the process of attaining the answers from the input images and questions. Phase 1 (black-colored paths) corresponds to the training of our OCCAM, in which we train the object-level feature extractor, the compositional reasoning module and the question embedding LSTM. Phase 2 (red-colored paths) corresponds to the induction of symbolic concepts based on the aforementioned trained neural modules, as well as the training of a concept projection module so that the induced concepts can be accommodated in the OCCAM pipeline. The figure shows the central role that the OCCAM plays in our framework.

3.2.1 Background on compositional reasoning

Notations. As shown in Figure 3.2, we name the visual vectors as vs, the output memory vector from the compositional reasoning module as m, the embedded word vectors for questions as ws, and the question embedding as q.

The compositional reasoning framework follows a VQA setting: given a question and an image as inputs, the model is required to return the correct answer choice. The target function can thus be written as:

$$\begin{aligned} \mathfrak{L}(ws, vs, q) = -\sum_{k \in K} y_k \log \mathcal{F}(q_k, \mathcal{G}(ws_k, vs_k, q_k)) \\ q_k = \mathcal{Q}(ws_k), vs_k = \mathcal{I}(im_k). \end{aligned} \tag{3.1}$$

K is the total number of image-question pairs, y is the one-hot ground truth vector, $\mathcal{F}$ is the classifier, $\mathcal{G}$ is the reasoning module, $\mathcal{Q}$ is the question embedding LSTM, $\mathcal{I}$ is the visual feature extractor and im is the image input.

The MAC reasoning module [5] processes visual and language inputs in a sequential way. Shown in Figure 3.2 (right), each MAC cell contains a control unit that uses word embedding to control what object features should be read and written into memory and an R/W (Read/Write) unit that performs reading and writing object features; the blue diagrams labeled with w stand for fully connected layers and the symbol $\odot$ stands for Hadamard product. More details can be found in Appendix A.1.

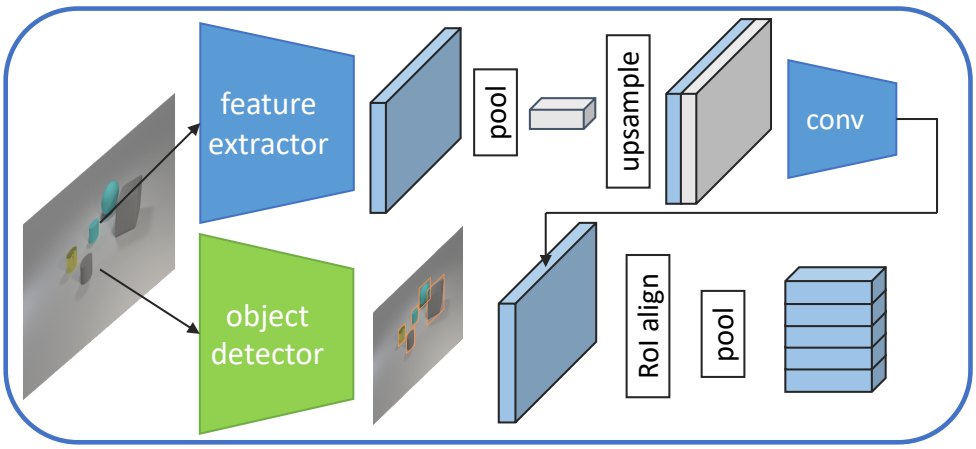


Figure 3.3: The architecture of our **object-level feature extractor**.

3.2.2 Object-centric compositional attention model

Our OCCAM network is shown in Figure 3.2 with phase 1 path. It performs MAC-style reasoning, but over the object-level visual features generated by our proposed **object-level feature extractor** (Figure 3.3). Fed with an image, the extractor produces a set of vectors vs, each encodes a single object's unary visual features and its interactions with other objects. The module works as the following steps:

(1) Following [12], we use Mask-RCNN [77] to detect all objects in an image and output the bounding boxes for them. The image is fed to a ResNet34 network [76] pretrained on ImageNet [78] to generate the feature maps.

(2) On top of the ResNet34 feature maps, we apply a global average pooling to get a single **global feature vector** (the gray vector in the figure). We concatenate this global vector with the feature map at each location, followed by three convolution layers. This global vector is crucial since it allows the visual features to encode the interaction among objects; and the three convolution layers fuse the local and global features into a single visual vector at each position.

(3) Finally, to get object-level features from the above pixel-level fused features, we use RoI align [77] to project the objects' bounding boxes onto the fused feature vectors to generate the RoI feature maps and average pool these RoI maps for each object to produce the object-level vs.

Our object feature extractor is jointly optimized with the reasoning module with Eqn (3.1) in the phase 1 training.

3.3 Concept Induction and Reasoning

This section describes how we achieve our goal of inducing symbolic concepts for objects and performing compositional reasoning on the induced concepts. First, we formalize the problem of concept induction (section 3.3.1). Second, built on the learned OCCAM network introduced in the previous section, we propose to induce concepts of both unary object properties or the binary relations between objects (section 3.3.2). Finally, we propose compositional reasoning over symbolic concepts by substituting the object-level features with the induced concepts (section 3.3.3).

3.3.1 Problem definition

We consider identifying three types of concepts: (1) the **unary concepts** $\mathcal{C}^u$ that are properties of objects (e.g., *red*, *cube*, etc.); (2) the **binary concepts** $\mathcal{C}^b$ that are relation descriptions between any two objects (e.g., *left*, *front* etc.); and (3) the **super concepts** $\mathcal{C}^{sup}$ that are hypernyms of certain subsets of concepts (e.g., *color*, *shape*, etc.), subject to that each object can only possess one concept under each super concept, e.g., *cube* and *sphere*.

As questions refer to objects and describe object relations in images and, more importantly, include all the semantic information to reach an answer, it is natural to induce the concepts from question words. Therefore, we assume that all the unary and binary concepts have their corresponding words; and these words are a subset of the nouns or adjectives from all the training questions. We denote the sets of words that describe unary concepts and binary concepts as M^u and M^b respectively. Therefore, the goal of concept induction consists of the following tasks:

• **Visual mapping:** for each concept $c \in \mathcal{C}^u$ or $\mathcal{C}^b$, learning a mapping from the visual feature v to c. In other words, a prediction function $f_c(v) \in \{0, 1\}$ is learned to predict the existence of concept c from the visual feature v of an object.

• **Word mapping:** for each concept $c \in \mathcal{C}^u$ or $\mathcal{C}^b$, identifying a subset of words $S_c \subset M^u$ or M^b that are synonyms representing the same concept, e.g., the concept of *'cube'* corresponds to set of words $\{cube, cubes, block, blocks, ...\}$.

• **Super concept induction:** clustering of concepts to form super concepts. Each super concept **c** contains a set of concepts $\{c_1, \cdots, c_k\} \subset \mathcal{C}^u$ or $\mathcal{C}^b$.

Algorithm 1: Classifier training data generation. ST(·) splits a vector $\alpha \in \mathbb{R}^{\beta}$ to a set of β values. GMM(·) uses Gaussian Mixture Model to cluster a set of data points. FB(·) finds the decision boundary for the 2 Gaussian components. $\mathbb{1}$ is the indicator function.

Result: P^u, P^b
$P^u = \{\}$, $P^b = \{\}$
for $x \in M^u \cup M^b$ **do**
  $S_x = \{\}$, $bd_x = 0$
for $vs, ws \in$ *DATASET* **do**
  for $c_w \in ws \cap M^u$ **do**
    $S_{c_w} = S_{c_w} \cup \text{ST}(\mathcal{R}(vs, c_w, m_0))$
  for $c_w \in ws \cap M^b$ **do**
    for $v \in vs$ **do**
      $S_{c_w} = S_{c_w} \cup \text{ST}(\mathcal{R}(vs, c_w, \mathcal{W}(m_0, v)))$
for $x \in M^u \cup M^b$ **do**
  $bd_x = \text{FB}(\text{GMM}(S_x))$
for $vs, ws \in$ *DATASET* **do**
  for $v_1 \in vs$ **do**
    for $c_w \in ws \cap M^u$ **do**
      $y = \mathbb{1}(\mathcal{R}(v_1, c_w, m_0) > bd_{c_w})$
      $P^u = P^u \cup \{(v_1, c_w, y)\}$
    for $c_w \in ws \cap M^b$ **do**
      for $v_2 \in \{vs - v_1\}$ **do**
        $y = \mathbb{1}(\mathcal{R}(v_1, c_w, \mathcal{W}(m_0, v_2)) > bd_{c_w})$
        $P^b = P^b \cup \{(v_1, v_2, c_w, y)\}$

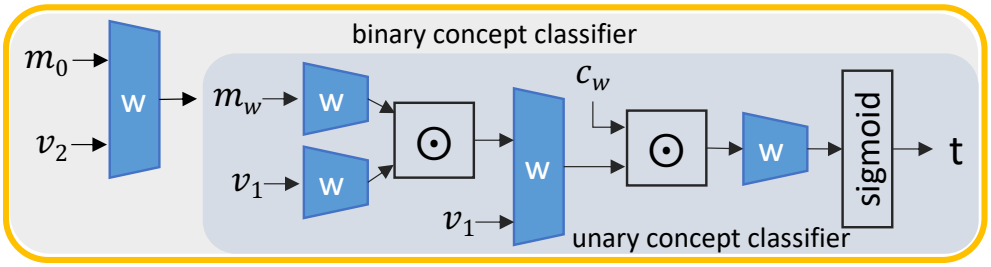


Figure 3.4: The structure of the **concept regression module**. v_1 and v_2 are the object-level visual vectors representing two objects respectively, and c_w is the word vector. m_0 is a fixed vector and m_w equals to m_0 for the unary concept classifier.

3.3.2 Concept induction

This section describes how we achieve the aforementioned tasks of concept induction. The key idea of our approach includes: (1) benefiting from the R/W unit from the trained MAC cells to achieve the visual mapping to textual words; (2) utilizing the inclusiveness of words' visual mapping to induce each concept's multiple word descriptions; and (3) clustering super concepts from the mutual

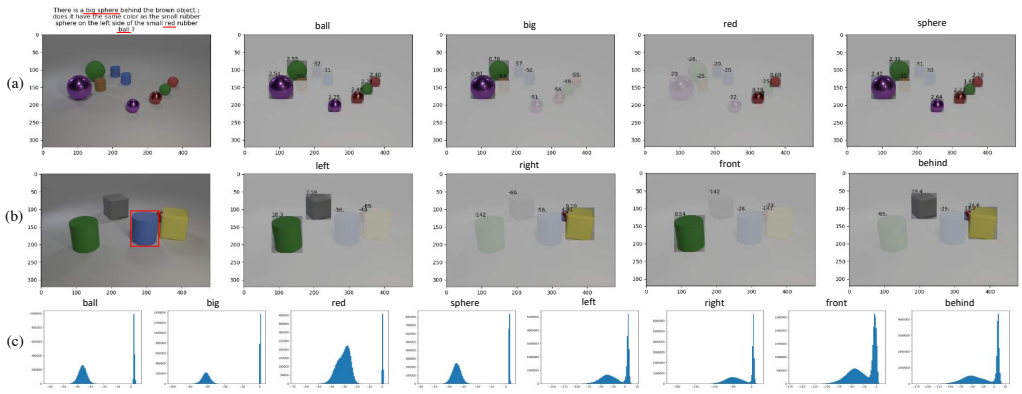


Figure 3.5: Attention visualization and attention logit distributions. (a) The attention visualization corresponding to the words describing the unary concepts by performing $\mathcal{R}(vs, c_w, m_0)$. Each of the words above the latter 4 images corresponds to a unique c_w and the value on each object is the attention logit (the same applies to (b)). (b) The attention visualization corresponding to the words describing the binary concepts by performing $\mathcal{R}(vs, c_w, \mathcal{W}(m_0, v_2))$. v_2 represents the object bounded by a red rectangle in the first image. (c) the attention logit distribution corresponding to each word describing a concept.

exclusiveness between concepts. To achieve the above, we first train two binary classifiers that can determine if a word correctly describes an object's unique feature and if a word correctly describes a relation between two objects respectively. Then, with the help of these classifiers, we produce zero-one vectors for words that properly describe the unique features for each object and the relations between any pair of objects in single images across the dataset. Finally, we perform a clustering method on the word vectors to generalize unary and binary concepts, and the super concept sets.

Visual mapping via regression from MAC cells The concept regression module is shown in Figure 3.4. It is composed of a classifier for the unary concept word regression, $\mathcal{B}^u(v_1, c_w) \in [0, 1]$, and a classifier for the binary concept word regression, $\mathcal{B}^b(v_1, v_2, c_w) \in [0, 1]$. $\mathcal{B}^u$ is expected to produce 1 if v_1 can be described by the word vector c_w. Likewise, $\mathcal{B}^b$ is expected to produce 1 if the relation of v_1 to v_2 can be described by the word vector c_w .

We generate training data points $P^u = \{({v_1}_i^u, {c_w}_i^u, y_i^u)\}$ and $P^b = \{({v_1}_i^b, {v_2}_i^b, {c_w}_i^b, y_i^b)\}$ for $\mathcal{B}^u$ and $\mathcal{B}^b$ by utilizing the Read/Write unit (Figure 3.2(right)) in the reasoning module after phase 1 training. The whole generation process is described in Algorithm 1. We denote $\mathcal{R}(vs, c_i, m_{i-1}) \in \mathbb{R}^{|O|}$ for the sequence of functions before the softmax operation in the Read unit and $\mathcal{W}(m_{i-1}, r_i) \in \mathbb{R}^D$ for the function

of the Write unit, where O is the set of objects in an image and D is the vector dimension.

Specifically, our algorithm first uses $\mathcal{R}(\cdot,\cdot,\cdot)$ and $\mathcal{W}(\cdot,\cdot)$ to find the attention logits on the objects corresponding to words describing the unary and binary concepts in a question as shown in Figure 3.5(a&b). We then use the values of logits to determine if the object possesses the concept of the word (positive) or not (negative). Noticing the attention logit distribution of the sampled objects for each word is a two-peak distribution (Figure 3.5(c)), we use a GMM [79] with two Gaussian components to model the distribution and find the decision boundary for each word's attention logit distribution. Observe that the distribution of a binary concept word has two interfering waves, because in some cases it is hard to tell if two objects have that relation ('front' is inappropriate if two objects are on the same horizon). P^u and P^b are generated by classifying the data points to positives and negatives with the decision boundaries. Finally, we can train $\mathcal{B}^u$ and $\mathcal{B}^b$ with data P^u and P^b by minimizing the binary cross entropy loss.

Binary coding of objects With trained $\mathcal{B}^u$ and $\mathcal{B}^b$, we represent an object o_1 with a binary code vector. Each dimension corresponds to a word. A dimension has value 1 if the corresponding word can describe o_1 and 0 otherwise. The binary vectors of object properties and of the relations between two objects, o_1 and o_2 can be computed with the functions $\gamma^u \in \mathbb{R}^{|M^u|}$ and $\gamma^b \in \mathbb{R}^{|M^b|}$ respectively:

$$\begin{aligned} \gamma^u &= \mathbb{1}_{i>0.5}(\mathcal{B}^u(v_1, C^u)) \\ \gamma^b &= \mathbb{1}_{i>0.5}(\mathcal{B}^b(v_1, v_2, C^b)), \end{aligned} \tag{3.2}$$

where v_1 and v_2 are the object-level visual vectors of o_1 and o_2, $C^u \in \mathbb{R}^{|M^u|\times D}$ and $C^b \in \mathbb{R}^{|M^b|\times D}$ are the stacks of word embeddings in vocabulary M^u and M^b. $\mathbb{1}_\alpha(\beta)$ performs elementwise on β: return 1 if the element satisfies condition α or 0 otherwise.

By applying γ^u and γ^b to all the objects and relations in the dataset, we can attain a matrix $\Gamma^u \in \{0,1\}^{M^u,N^u}$ and a matrix $\Gamma^b \in \{0,1\}^{M^b,N^b}$ as shown in Figure 3.6, where N^u and N^b are the total numbers of objects and co-occurred object pairs. The two matrices summarize each word's corresponding visual objects in the whole dataset.

Concept/super-concept induction Finally, we group synonym words to unary and binary concepts and generate the super concepts. These two tasks are achieved

	small	tiny	big	cube	ball	⋯
	1	1	0	1	0	⋯
	0	0	1	1	0	⋯
	1	1	0	0	1	⋯
	0	0	1	0	1	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋱

(a)

	left	right	front	behind
	0	1	1	0
	1	0	0	1
	1	0	1	0
⋮	⋮	⋮	⋮	⋮

(b)

Figure 3.6: The zero-one matrices indicating word descriptions of objects and object relations across the dataset. (a) The matrix Γ^u indicates what words can describe objects. (b) The matrix Γ^b indicates what words can describe the relations object v_1's (bounded by green rectangles) are to object v_2's (bounded by red rectangles).

via exploring the word inclusiveness and the concept exclusiveness captured by Γ^u and Γ^b: (1) words describing the same concept correspond to similar column vectors, e.g., Γ^u_{small} and Γ^u_{tiny}; (2) words describing exclusive concepts have column vectors that usually do not have 1 values on same objects simultaneously, e.g., Γ^u_{cube} and Γ^u_{ball}. Based on the aforementioned ideas, we define the correlation metric between two words c_{w_1} and c_{w_2} as below:

$$\theta_{c_{w_1},c_{w_2}} = P(\gamma_{c_{w_1}} = 1 \mid \gamma_{c_{w_2}} = 1) + P(\gamma_{c_{w_2}} = 1 \mid \gamma_{c_{w_1}} = 1) \tag{3.3}$$

$$= \frac{|\Gamma_{c_{w_1}} \odot \Gamma_{c_{w_2}}|^1_1}{|\Gamma_{c_{w_2}}|^1_1} + \frac{|\Gamma_{c_{w_1}} \odot \Gamma_{c_{w_2}}|^1_1}{|\Gamma_{c_{w_1}}|^1_1}. \tag{3.4}$$

This guarantees that $\theta \to 0^+$ for two synonym words, $\theta \to 2^-$ for two words corresponding to exclusive concepts and $\theta \in (0, 2)$ for words corresponding to different nonexclusive concepts. We can produce the correlation sets for the words describing the unary concepts and the binary concepts respectively with Eqn (3.5).

$$\Theta^x = \{\theta_{c_{w_1},c_{w_2}}\}; c_{w_1}, c_{w_2} \in M^x; x \in \{u, b\} \tag{3.5}$$

Our final step fits two GMM on Θ^u and Θ^b respectively. Each GMM has three components $\mathcal{N}_0$, $\mathcal{N}_1$ and $\mathcal{N}_2$, with their mean values initialized with 0,1 and 2. We then induce the unary and binary concepts, where each concept consists of synonym words whose mutual correlation is clustered to the Gaussian component $\mathcal{N}_0$. Similarly, we induce the super concepts, where each super concept contains multiple concepts and any two words from different concepts have correlation

Algorithm 2: Concept vector generalization. MAX(α) and HARDMAX(α) return the largest value in vector α and its position as a one-hot vector, respectively.

Result: K^u, K^b
$K^u = \mathbf{0}^{|O|\times|E^u|}$, $K^b = \mathbf{0}^{|O|\times|O|\times|E^b|}$
for $i \in O$ **do**
  for $e^u \in E^u$ **do**
    $K^u[i][e^u]$ =MAX($\mathcal{B}^u(v_i, \rho_{e^u})$)
  for $l^u \in L^u$ **do**
    $K^u[i][l^u]$ =HARDMAX($K^u[i][l^u]$)
  for $j \in O - \{i\}$ **do**
    for $e^b \in E^b$ **do**
      $K^b[i][j][e^b]$ =MAX($\mathcal{B}^b(v_i, v_j, \rho_{e^b})$)
    for $l^b \in L^b$ **do**
      $K^b[i][j][l^b]$ =HARDMAX($K^u[i][j][l^b]$)

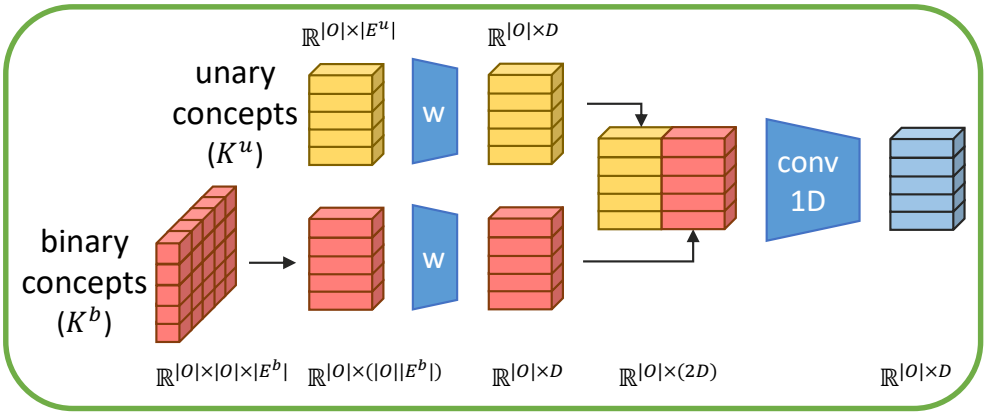


Figure 3.7: The structure of the **concept projection module**. We label the dimensions of matrices near them in the graph.

clustered to the Gaussian component of $\mathcal{N}_2$.

We denote the set of words corresponding to a concept e as ρ_e, the set of the super concept sets as L, the set of all concepts as E. Then, we can represent all the objects in an image with a unary concept matrix K^u and represent all the relations between any two objects in an image with a binary concept matrix K^b with Algorithm 2.

3.3.3 Concept compositional reasoning

Our ultimate goal is to perform compositional reasoning to answer a question with the generated concept representations K^u and K^b for an image; so as to

Table 3.1: The comparison of our OCCAM framework to the state-of-the-art methods on CLEVR (up) and GQA (down) datasets. † means training with additional program supervision. ‡ means pretraining on larger visual+language corpora. § means pretraining a scene-graph extraction model with additional rich annotated data.

(a) CLEVR

method	overall	count	exist	comp num	query attr	comp attr
RN [80]	95.5	90.1	93.6	97.8	97.1	97.9
FiLM [4]	97.6	94.5	93.8	99.2	99.2	99.0
MAC [5]	98.9	97.2	99.4	**99.5**	99.3	99.5
NS-CL [12]	98.9	**98.2**	99.0	98.8	99.3	99.1
OCCAM (ours)	**99.4**	98.1	**99.8**	99.0	**99.9**	**99.9**
NS-VQA† [11]	99.8	99.7	99.9	99.9	99.8	99.8
Human [9]	92.6	86.7	96.6	86.5	95.0	96.0

(b) GQA

method	val	test-dev	test
MAC [7]	57.5	-	54.1
LXMERT [81]	-	50.0	-
LCGN [82]	63.9	55.8	56.1
OCCAM (ours)	**64.5**	**56.2**	**56.2**
MMN [83]†	-	60.4	60.8
NSM [84]§	-	63.0	63.2
LXMERT [81]‡	-	60.0	60.3
ViLT [85]‡§	-	65.1	64.7

confirm that our induced concepts are accurate and sufficient. We achieve this with the phase 2 training process in Figure 3.2. The key idea is to transplant the learned compositional reasoning module from manipulating the visual features to manipulating K^u and K^b, for attaining the answer to a question.

To this end, first, we project K^u and K^b to the same vector space with vs with the concept projection module shown in Figure 3.7, so that the compositional module can perform the reasoning steps on the projected concept vectors. Specifically, we first reduce the dimension of K^b from $\mathbb{R}^{|O|\times|O|\times|E^b|}$ to $\mathbb{R}^{|O|\times|O||E^b|}$, resulted in $\hat{K}^b$, because K^b can be understood as the relations to other objects for each object in an image. Then, we use two separate fully connected networks to project K^u and $\hat{K}^b$ respectively, concatenate and use a sequence of 1D convolution layers to project the results to the same dimension of vs's.

Second, to minimize the discrepancy between the distribution of our projected vectors and that of the original visual vectors vs, we fix the weights of other mod-

ules in the framework and only train the concept project module by optimizing the target function Eqn. (3.1). Then, we train the concept projection module and the compositional reasoning module with other modules' weights fixed to better optimize Eqn. (3.1). The result is a compositional reasoning model that works on the induced concepts only.

3.4 Experiments

3.4.1 Settings

Datasets (1) We first evaluate our model on the **CLEVR** [86] dataset. The dataset comprises images of synthetic objects of various shapes, colors, sizes and materials and question/answer pairs about these images. The questions require multi-hop reasoning, such as finding the transitive relations, counting numbers, comparing properties, to attain correct answers. Each question corresponds to a ground truth human-written programs. Because the programs rely on predefined concepts thus do not fit our problem, we let our framework learn from scratch *without* using the program annotations. There are 70k/15k images and $\sim$700k/$\sim$150k questions in the training/validation sets. We follow the previous works [11, 5, 12] to train our model on the whole training set and test on the validation set.

(2) To demonstrate the generalizability of our approach, we further evaluate on the **GQA** dataset. GQA is a real-world visual reasoning benchmark. It consists of 113K images collected from the Visual Genome dataset [87] and 22M questions. It has a train split for model training and three test splits (val, test, test-dev) [88]. The dataset provides the detected object features extracted from a Faster RCNN detector [89], so each object is represented as a 2048-dimensional vector.

Implementation details We include the checklist of our implementation details in Appendix A.2.

3.4.2 Object-level reasoning

We first perform the end-to-end phase 1 training shown in Figure 3.2, i.e., our OCCAM model. The performance comparison of our model to the state-of-the-

Table 3.2: Effect of the choice of reasoning steps for our model.

steps	4	8	12	16
accuracy (CLEVR)	94.3	98.6	**99.4**	99.1
accuracy (GQA test-dev)	55.1	55.6	55.2	**56.2**

art models is shown in Table 3.1. Under the setting that no external human-labeled programs and no pretraining are used, our model achieves state-of-the-arts compared with published results on both CLEVR and GQA datasets. For comparison with models on GQA leaderboard, we also train our OCCAM model on train-all split and achieves an accuracy of 58.5% on the test-standard split of GQA dataset, which outperforms other popular models (e.g. MCAN, BAN and LCGN) trained with no additional data (accuracies are 57%$\sim$58%). While transformer-based methods with pretraining phase boost the performance, however, they undermine the model's explainability and make it difficult to induce concepts. On CLEVR, our model also has an on-par performance with the best model [11] that uses external human-labeled programs.

Compared to the original MAC [5] framework which uses image-level attentions, our model proves that the constraint of attentions on the objects are useful for improving the performance on both datasets, with significant improvement on the validation sets. We do not use the position embedding to explicitly encode the positions of objects for relational reasoning; however, we use the global features to enhance the model's understanding of inter-object relations. This shows that the relations among objects are learnable concepts without external knowledge for the deep network.

Table 3.2 further gives an ablation study on the numbers of reasoning steps, i.e., the number of MAC modules, for our model. The reasoning model with 4 steps has a performance gap to the models with 8, 12 or 16 steps, while the latter three models have on-par performances. We conjecture that the model with low reasoning steps may not be able to capture multiple hops of a question and the model performance converges with an increasing number of reasoning steps. We also did an ablation study on the contribution of object-level feature extractor on CLEVR dataset. With pretrained ResNet101 features, learnable ResNet34 features, learnable ResNet34 features plus global features respectively, the model achieves 97.9%, 99.0% and 99.4% on the validation set. It shows the importance of enhancing global context understanding at object level.

Table 3.3: Comparison of our visual feature-based OCCAM and our concept-only OCCAM. The number of reasoning steps is 8.

(a) CLEVR

method	overall	count	exist	comp num	query attr	comp attr
$\text{OCCAM}_{\text{visual}}$	98.6	95.9	99.8	96.2	99.8	99.7
$\text{OCCAM}_{\text{concept}}$	97.9	95.6	98.7	97.3	98.4	99.3

(b) GQA

method	val	test-dev
$\text{OCCAM}_{\text{visual}}$	63.8	55.6
$\text{OCCAM}_{\text{concept}}$	63.1	54.2

3.4.3 Concept induction and reasoning

Next, we evaluate the performance of our concept induction method, i.e., the phase 2 training in Figure 3.2. To qualitatively show that our induced concepts capture sufficient and accurate information for visual reasoning, we replace the visual inputs to the objects' induced concepts according to Section 3.3.3. The resulted model, denoted as $\text{OCCAM}_{\text{concept}}$, is expected to perform closely to the original OCCAM with high-quality induced concepts.

Table 3.3 gives the results. To achieve the balance of the performance and the interpretability, we make the OCCAM model run 8 reasoning steps for both concept induction and reasoning. It is observed that our concept-based OCCAM (with induced concept features only) achieves on-par performance with the original OCCAM model (with full input visual features). We also visualize the reasoning steps for the $\text{OCCAM}_{\text{concept}}$ model in Appendix A.6.

Human study of concept induction We present the unary concept correlations Θ^u in Figure 3.8 and 3.9 for both CLEVR and GQA. Since GQA consists of a huge vocabulary with highly-correlated concepts, we demonstrated a sub-set of concepts associated to general words/phrases.

On CLEVR, the concept definition from the data generator can be perfectly recovered by our approach: from Figure 3.8, the correlation between any pair of synonyms is close to 2, the correlation between words belonging to the same super concept set is close to 0, and the correlation between words belonging to two different super concept sets is in the middle of the range$[0, 2]$. Appendix A.3 provides the full generated concept hierarchy, which perfectly matches the definition in CLEVR generator and human prior knowledge, i.e., 100% accuracy, according

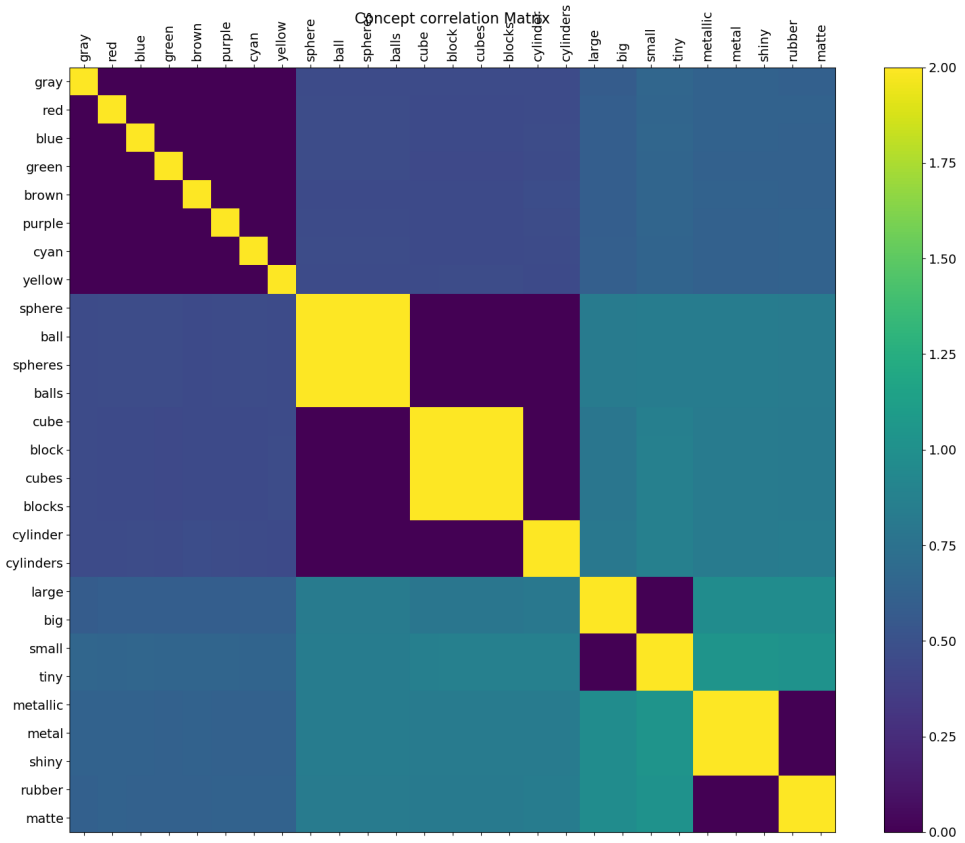


Figure 3.8: The CLEVR unary concept correlations Θ^u.

to our human investigation.

On GQA, the correlations between words are complicated. We present a subset of word correlations in Figure 3.9. Instead of using Eqn. (3.4), here we show the conditional probability that a column attribute exists given that the row attribute exists. We observe that words describing a similar property have a high positive correlation, such as 'yellow' and 'orange', 'concrete' and 'stone'. They can be grouped into a single concept. Words of exclusive meanings negatively correlates with each other, such as 'flying' and 'standing', 'pointy' and 'sandy'. They can be grouped into a super concept. However, the real-world data makes it difficult to induce some commonsense super concepts. For example, the same object can have multiple colors (e.g., sky can be both gray and blue). Also, object concept can have degrees (light or dark blue), so we have to use soft values to represent the concepts. We additionally conduct human studies on the pairwise accuracy of detected concept and super concept clusters, which can be found in Appendix A.7.

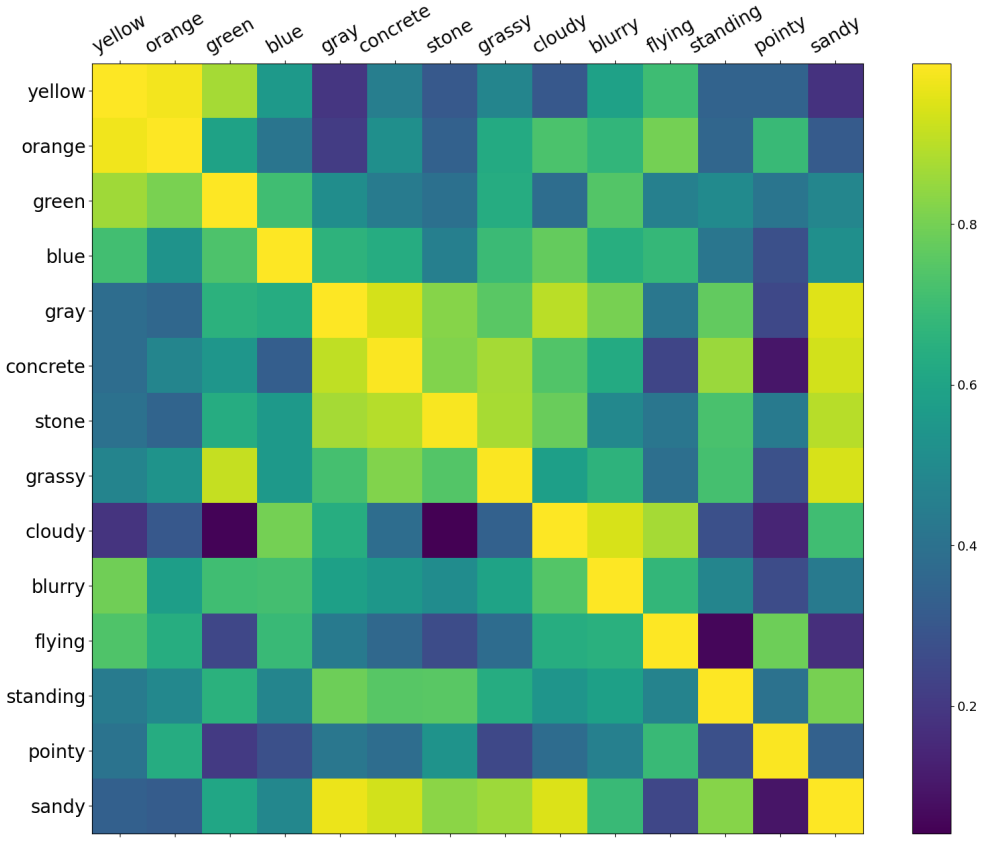


Figure 3.9: The subset of GQA concept correlations.

We provide more visualization results in Appendix A.4 and A.5, including the extension of word analogy [90] (e.g., "Madrid" - "Spain" + "France" $\rightarrow$ "Paris") to multi-modality and the quantification of distance between two visual objects with the super concept space.

3.5 Conclusions

Our proposed OCCAM framework performs pure object-level reasoning and achieves a new state-of-the-art without human-annotated functional programs on the CLEVR dataset. Our framework makes the object-word cooccurrence information available, which enables induction of the concepts and super concepts based on the inclusiveness and the mutual exclusiveness of words' visual mappings. When working on concepts instead of visual features, OCCAM achieves comparable performance, proving the accuracy and sufficiency of the induced concepts. For future works, our method can be extended to more sophisticated induction tasks,

such as inducing concepts from phrases, with more complicated hierarchy, with degrees of features (e.g., *dark blue*, *light blue*) and inducing complicated relations between objects (e.g. *a little bigger*).

CHAPTER 4

UNLOC: A UNIFIED FRAMEWORK FOR VIDEO LOCALIZATION TASKS

4.1 Introduction

Contrastive vision-language pretraining has been shown to learn powerful feature representations, and, moreover, enables open-set inference on a wide range of tasks [91, 92]. As a result, pretrained models such as CLIP [91] have been adapted to multiple diverse tasks including video classification [93, 94], object detection [95] and segmentation [96].

In this work, we study how to adapt large-scale, contrastively trained image-text models to untrimmed video understanding tasks that involve localization. While CLIP has been used widely for trimmed video tasks (classification [93, 94] or retrieval [97]), its use on long, untrimmed video is still in a nascent stage. Long videos come with multiple challenges - CLIP is pretrained on images only, and localization in untrimmed videos requires exploiting fine-grained temporal structured information in videos. In particular, it is challenging for image and language models to learn properties of temporal backgrounds (with respective to foreground actions) during training. In contrast, natural videos often come with a large, variable proportion of background and detecting specific actions is critical for localization tasks [38]. Finally, localization in long untrimmed videos also typically involves detecting events at multiple temporal scales. Consequently, existing approaches that use CLIP typically focus on a two-stage approach involving off-the-shelf proposal generators [37], or use temporal features such as I3D [47] or C3D [98]. In contrast, we propose an end-to-end trainable one-stage approach starting from a CLIP two-tower model only.

We focus specifically on three different video localization tasks - Moment Retrieval (MR) [99, 40], Temporal Action Localization (TAL) [100, 101] and Action Segmentation (AS) [52]. These tasks have typically been studied separately, with different techniques proposed for each task. We show how we can use a single,

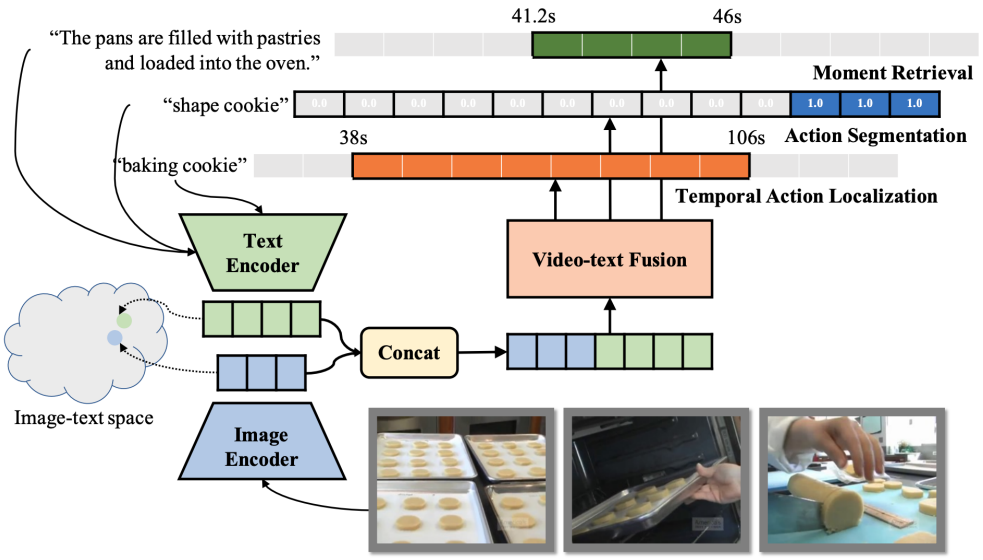


Figure 4.1: Applying two-tower CLIP to video localization tasks: We propose UnLoc, a single stage, unified model that achieves state of the art results on 3 different video localization tasks - moment retrieval, temporal action localization and action segmentation. UnLoc leverages a two-tower model (with a vision and text encoder) in conjunction with a video-text fusion module and feature pyramid to perform mid-level feature fusion without the need for any temporal proposals.

unified approach, to address all of these tasks, without using any external proposals. We do this by leveraging a two-tower model (with a vision and text encoder), in conjunction with a single video-text fusion module, which performs mid-level fusion of text and visual tokens (Fig. 4.1). Our two-tower model can naturally handle tasks such as moment retrieval which contain both video and text as input modalities, and can be used for open-set inference in other tasks such as temporal action localization and action segmentation. While many works use the visual encoder only [33, 29, 102, 36], we believe that the language priors learnt with the pretrained text encoder can contain useful information and should be leveraged together with the image encoder early in the model design (particularly for open-set evaluation), and not right at the end for similarity computation. Inspired by existing object detection works [103], we also use the output frame tokens from our fusion module to construct a feature pyramid, to enable understanding at multiple temporal scales.

Our approach achieves state-of-the-art results across all three video localization tasks - MR [99, 40], TAL [100, 101] and AS [52]. We also perform thorough ablation studies, studying the effect of modeling choices across a range of tasks.

4.2 Methods

Our model unifies three tasks: MR, TAL and AS, which we first define in Sec. 4.2.1. As shown in Fig. 4.2, our model (Sec. 4.2.2) first tokenizes a (video, text) pair and then fuses information from the two modalities together with a simple video-text fusion module. To capture the multi-scale information needed for localization, we then construct a Feature Pyramid (Sec. 4.2.3) on the output of the video-text fusion module. These multi-scale features are then fed into a task-specifc Head module (Sec. 4.2.4) to localize activities or "ground" a language description.

4.2.1 Tasks

Moment Retrieval (MR), also known as Video Grounding, is the task of matching a given language description (query) to specific video segments in an untrimmed video. Temporal Action Localization (TAL) aims to detect events in a video and output the corresponding start- and end- timestamps. One key difference from MR is that events in TAL are from a predefined closed-vocabulary set, often described by a short phrase (e.g., "baking cookies"). Finally, similar to Semantic Segmentation, which parses images into semantic categories at a pixel level, Action Segmentation (AS) involves producing activity labels at a frame level. Also, for this task the labels are typically predefined from a closed-vocabulary set.

4.2.2 A unified architecture

Our model takes (video, text) pairs as inputs, and for each frame in the video it outputs a relevancy score between the frame and the input text, as well as the time differences between the frame and the start/end timestamps of the predicted segment. The target relevancy score is set to 1 if a frame falls within the labeled segment, otherwise zero. In the case of TAL and AS, we use class labels as the input texts while in MR, text queries are used as input texts. For each video we form C (video, text) pairs where C is the number of classes in TAL and AS and for MR C is the number of captions associated with this video.

Fig. 4.2 gives an overview of our proposed architecture. The input pair is first tokenized and encoded by a pair of image and text encoders. The two encoders are

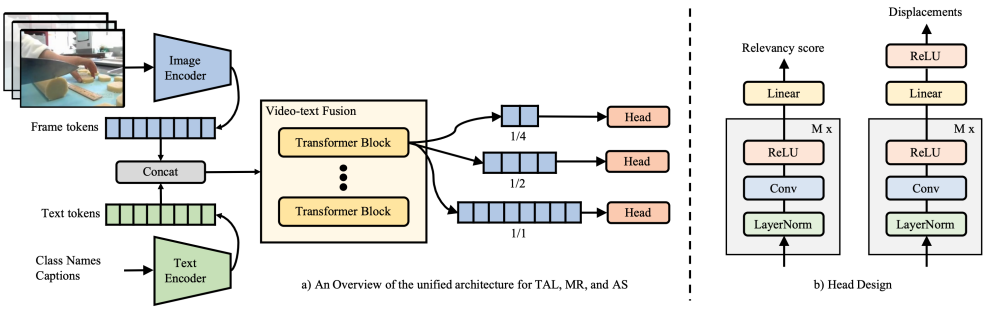


Figure 4.2: Overview of our method UnLoc. Given a video and text (e.g., class names in TAL/AS or captions in MR) pair, first they are tokenized and encoded by a pair of image and text encoders. Frame and text tokens are concatenated into a long sequence and then fed into a transformer for fusion. Frame tokens from the last transformer layer are used to construct a feature pyramid in which each level connects to a head to predict a per-frame relevancy score and start/end time displacements. No text token is used to construct the feature pyramid since text information has already been "fused" into the frame tokens via self-attention. We show a 3 layer feature pyramid for simplicity. All heads across different pyramid levels share the same weights.

initialized from a pretrained, CLIP visual language model [91]. The two encoders come from the same pretrained model - pretrained by aligning image and text pairs with a contrastive loss, which provides a strong prior on measuring the relevancy between each frame and the input text. This is one of key contributing factors to the success of our model. As Sec. 4.3.2 will show, using "unpaired" image/text encoders indeed diminishes the performance.

After tokenization and encoding, the input video and text are represented by N frame tokens and T text tokens. We then form a new sequence by concatenating N frame tokens with either a single token (e.g., CLS) representing the whole text sequence or all T text tokens (See Sec. 4.3.2 for ablation). The concatenated sequence is then fed into a video-text fusion module. In this work, we implement this fusion module using a transformer encoder [104, 105]. This encoder performs two key functions - (i) it is a temporal encoder, able to model inter-frame correspondences omitted by the image-only CLIP model, and (ii) it can also function as a refinement network, with the ability to correct mistakes made by the CLIP model. After fusion, only frame tokens $\mathbf{X}^c \in \mathbb{R}^{N \times K}$ are used to construct a feature pyramid where each level is created by downsampling the original sequence using strided convolutions where c is the index of the class or caption and K denotes the hidden size of the token. This process is repeated for all class labels/captions. Text tokens are omitted from this construction because their information has been incorporated into the frame tokens by the fusion module, and

they do not correspond to any timestamps.

Finally, each pyramid level connects to a Head module to predict a per-frame relevancy score, $\hat{\mathbf{y}}_l^c \in \mathbb{R}^{N_l}$, and start/end time displacements, $\hat{\mathbf{t}}_l^c \in \mathbb{R}^{N_l \times 2}$, where N_l denotes the number of features in pyramid level l. The final number of predictions is $\sum_{i=1}^{L} \frac{N}{2^{i-1}}$, and is, therefore, greater than N if there is more than one level in the feature pyramid. For example, if we construct a 3-level feature pyramid the total number of predictions will be $N + N/2 + N/4$. Each prediction is expanded into a temporal segment by applying the predicted displacements to its frame timestamp. Given these temporal segments for all pyramid layers, we filter out overlapping segments during inference with soft non-maximal suppression (SoftNMS) [106].

4.2.3 Feature pyramid

A feature pyramid can improve a model's capability to detect events at different scales. For example, features from the top level can detect events with a long duration while bottom-level features can localize short segments. Feature Pyramid Networks (FPN [107]) have been used extensively in object detection for images to pass richer semantic information from a higher level in the CNNs to lower level feature maps that have higher spatial resolution. We propose another simpler structure inspired by ViTDet [103] by removing the lateral and top-down connections in the FPN. Since the last layer in the transformer encoder contains the most semantic information [108] and shares the same temporal resolution as the first one, the lateral and top-down connections are no longer required. The feature pyramid is constructed by applying convolution with different strides to the output tokens from the last transformer layer in the video-text fusion module (See Fig. 4.2a). Note that text tokens are not used during the feature pyramid construction, since their information has been fused into the frame tokens. This simpler design removes the downsampling step in the encoder and allows us to share the same architecture used in pretraining stage (See Sec. 4.3.1 for more details). Similar to findings in [103], our ablation (Sec. 4.3.2) shows that this simpler design outperforms FPN on TAL, as it introduces less additional layers to the pretrained model. AS is a frame-level task, so features from only the bottom level in the feature pyramid are used for prediction.

4.2.4 Head design

As shown in Fig. 4.2b, we have two heads, one for relevancy score prediction and the other for displacement regression. Although the two heads share the same structure, their weights are not shared. Our head design following [50] consists of M 1D convolution blocks where each block is made of three operations: Layer Normalization [109], 1D convolution, and a ReLU activation [110]. A convolution (e.g., a local operation) is used to encourage nearby frames to share the same label. At the end of each head, a linear layer is learned to predict per-frame relevancy scores $\hat{\mathbf{y}}^c \in \mathbb{R}^{N\times 1}$ or to predict per-frame start/end time displacements $\Delta\hat{\mathbf{t}}^c \in \mathbb{R}^{N\times 2}$:

$$\hat{\mathbf{y}}^c = \mathbf{Z}^c\mathbf{w}_{cls} + b_{cls} \tag{4.1}$$

$$\Delta\hat{\mathbf{t}}^c = \text{relu}\left(\mathbf{Z}^c\mathbf{w}_{reg} + \mathbf{b}_{\text{reg}}\right) \tag{4.2}$$

where $\mathbf{Z}^c$ are the activations of frame tokens $\mathbf{X}^c$ after convolution blocks, $\mathbf{w}_{cls} \in \mathbb{R}^{K\times 1}$ and $b_{cls} \in \mathbb{R}^{1\times 1}$ are the weights and bias for the classification head, and $\mathbf{w}_{\text{reg}} \in \mathbb{R}^{K\times 2}$ and $\mathbf{b}_{\text{reg}} \in \mathbb{R}^{1\times 2}$ are the weights and biases for the regression head. We limit the predicted displacements to be greater or equal to zero through a ReLU non-linearity. This process is repeated to generate scores and displacements for every class/caption and the same learned weight and bias terms are shared. For AS only the relevancy scoring head is used. One key difference from [36] is that our model predicts a different start/end time displacement for each class while [36] predicts one displacement $\Delta\hat{\mathbf{t}} \in \mathbb{R}^{N\times 2}$ shared among all classes, which assumes that there is no overlapping segment in the video.

4.2.5 Loss function

For AS, we use sigmoid cross entropy loss to measure the relevance between a frame and class label. For TAL and MR, we use the focal loss [111] for the relevancy scoring head as class imbalance is a known issue in one-stage detectors [111]. For the regression head we experiment with four popular regression losses, L1, IoU, DIoU [112], and L1+IoU. The L1 loss computes the absolute distance between the predicted and the ground truth start/end times.The IoU loss directly

optimizes the intersection of union objective, which is defined as

$$L_{iou} = 1 - \text{IoU}(\Delta\hat{s}, \Delta\hat{e}) \tag{4.3}$$

$$= 1 - \frac{\min(\Delta\hat{s}, \Delta s) + \min(\Delta\hat{e}, \Delta e)}{\max(\Delta\hat{s}, \Delta s) + \max(\Delta\hat{e}, \Delta e)} \tag{4.4}$$

where $\Delta\hat{s}, \Delta\hat{e}$ and $\Delta s, \Delta e$ are the predicted and the ground truth displacements to the start/end times. If $\Delta\hat{s}$ or $\Delta\hat{e}$ is zero, its gradient will also be zero, which could happen due to poor initialization. Distance IoU (DIoU [112]) is proposed to address the zero-gradient issue by also taking into account the distance between the two centers of the ground truth box and the predicted box. We end up using L1 loss based on the ablation in Sec. 4.3.2, and also apply a weight factor α to balance between the focal loss and L1 loss.

4.3 Experiments

We first describe datasets, evaluation metrics and implementation details in Sec. 4.3.1. We then provide a number of ablations on our architecture design, use of the text encoder, video-text fusion module and finetuning strategies (Sec. 4.3.2). Finally, we show the results of our method compared to the state-of-the-art in Sec 4.3.3.

4.3.1 Experimental setup

Datasets and evaluation metrics

Moment retrieval. ActivityNet Captions [99] contains 20,000 videos and 100,000 segments where each is annotated with a caption by human. On average, each caption contains 13.5 words and videos have an average duration of 2 minutes. The dataset is divided into three splits, train, val_1, and val_2. Following [45, 98], we use train split for training, val_1 for validation and val_2 for testing. Charades-STA [40] contains 6,672 videos and 16,128 segment/caption pairs, where 12,408 pairs are used for training and 3720 for testing. Each video is annotated with 2.4 segments on average and each has an average duration of 8.2 seconds. QVHighlights [48] includes over 10,148 cropped videos (150s long), and each video is

annotated with at least one query describing the relevant moments (24.6s in average). In total, there are 10,310 text queries with 18,367 associated moments. Following [48, 49], we use train split for training and val split for testing. The most commonly used metric for moment retrieval is the average recall at k computed under different temporal Intersection over Union (IoU) thresholds, which is defined as the percentage of at least one of the top-k predicted segments having a larger temporal IoU than the threshold with the ground truth segment, i.e. Recall $@\ \mathrm{K}, \mathrm{IoU} = [0.5, 0.7]$.

Temporal action localization. ActivityNet 1.3 [100] is a collection of 20,000 untrimmed videos focusing on human actions. Most videos contain only one labeled segment and segments in one video are from the same action class. The dataset is divided into three subsets, *train*, *validation*, and *test*. Following standard practice [31, 32, 113, 36], we train our models on the training set and report results on the validation set. The standard evaluation metric for temporal localization is mean Average Precision (mAP) computed under different temporal IoU thresholds. We report mAP under an IoU threshold of 0.5, denoted as mAP@ $0.5\mathrm{IoU}$. We also report results for the zero-shot setting, following the data split protocols proposed by [37, 38]: 1) training on 50% of the action labels and testing on the remaining 50%; 2) training on 75% of the labels and testing on the rest 25%. These are created using 10 random splits of the data, following [37, 38]. In the rest of contents, we use ANet TAL and ANet MR to denote ActivityNet 1.3 and ActivityNet Captions, respectively.

Action segmentation. The COIN [52] dataset consists of 11,827 training videos and 2,797 testing videos. Each video is labeled with an average of 3.9 segments where each segment lasts 14.9 seconds on average. The segment labels describe a step needed to complete a task, such as "take out the old bulb", "install the new bulb", etc. Frame accuracy is the primary metric used in the COIN action segmentation task, which is defined as the number of correctly predicted frames divided by the total number of frames. However, given how a large proportion of the frames are labelled as background (58.9%), a naive majority-class prediction model will already get an accuracy of 58.9%. Hence, we also report mean Average Precision (mAP), which averages AP over the classes (excluding background) and is therefore not directly impacted by the large proportion of background.

Implementation details

Model Architecture: In UnLoc-Base and Large models, the image and text encoders follow the same architecture used in CLIP-B and CLIP-L. The video-text fusion module is implemented using a 6-layer Transformer and the hidden size is set to 512 and 768 for UnLoc-B and UnLoc-L and the MLP dimension is set to 2048 and 3072, respectively, We construct a 4-layer feature pyramid from the last layer in the video-text fusion module following the procedure described in Sec. 4.2.3. Following [36], an output regression range is specified for each level in the pyramid, which is set to $[0, 4], [4, 8], [8, 16], [16,$ inf], respectively ordered from bottom to the top. All heads across different pyramid levels share the same weights, and are randomly initialized.

Pretraining: Our models are pretrained on Kinetics (K700 [114] for our best models, K400 for ablations). The pretraining task is a 400/700-way binary classification problem using a sigmoid cross entropy loss. For example, for each video we feed all class names into the text tower and the objective is to classify whether or not the video matches any of the class names. During Kinetics pretraining, the image encoder is finetuned and the text encoder is kept frozen to avoid catastrophic forgetting due to the fact that we are finetuning on a small fixed set of vocabulary in Kinetics. The video-text fusion module is always finetuned.

Training: In training the frames are first resized to have a shorter side of 256 and models are trained on a random crop of size 224×224. For TAL and AS class names are augmented using Kinetics prompts released by [91], e.g., "a video of a person doing {label}". Unless specified otherwise, all TAL and MR models are trained on 128 frames evenly spaced sampled across the whole video. This follows the sampling strategy adopted by [36] to deal with videos of varying lengths. Unless specified otherwise, for AS on the COIN dataset, we extract the RGB frames at 2 FPS, which is the labelling resolution. We randomly sample 512 consecutive frames and apply padding for videos with less than 512 frames. All models are trained using synchronous SGD with a momentum of 0.9 , with a batch size of 64 . We follow [115] and apply the same data augmentation and regularization schemes [116, 117], which were used by [118] to train vision transformers more effectively. For more implementation details and hyperparameters, we refer readers to the appendix and code.

Inference: During inference, our results are obtained evaluating a single central crop of 224×224. For AS on COIN, we run our model in a non-overlapping

Table 4.1: Effect of architecture design and losses. Results are presented on the ANet TAL for mAP@ 0.5IoU. We compare 4 popular regression losses, two types of feature pyramids (and no pyramid), and the number of convolutional layers in the localization heads.

Losses				Feature Pyramid			# conv layers			
L1	IoU	L1+IoU	DIoU	No	FPN	ViTDet	1	2	3	4
54.6	54.0	53.9	54.1	47.3	53.8	**54.7**	52.5	53.4	**54.7**	54.5

Table 4.2: Effect of different text encoders. We use the same frozen CLIP-B image encoder, with both T5 and CLIP-B text encoders and show results across all tasks. Paired image/text encoders significantly outperform unpaired encoders for localization tasks. Note that for COIN, results are reported using mAP.

Text Encoder	MParams	ANet TAL	ANet MR	COIN
T5-S	147.1	46.7	39.7	16.1
T5-B	221.5	46.6	39.9	15.9
CLIP-B	174.9	**53.3**	**44.2**	**16.4**

sliding window fashion with a window size of 512 frames. For TAL and AS, we report two results, one using the first prompt and the other by averaging all 28 context prompts, which is defined as prompt ensembling in [91].

4.3.2 Ablations

We use the hyperparameters described in Sec. 4.3.1 as the default setting for all experiments in the ablation unless specified otherwise. For AS on COIN we randomly sample 128 consecutive frames (instead of 512) for efficiency during training for the ablations. For the ablations we report ANet TAL with mAP@0.5IoU, ANet MR with Recall@ 1 under IoU $= 0.5$ and COIN with mAP.

Architectural design choices. In Table 4.1, we ablate three design choices: the loss function, feature pyramid design, and the number of convolution layers in the localization heads. All losses perform similarly with L1 being slightly better than other three. ViTDet-style feature pyramid outperforms a standard FPN [107] as it introduces less additional layers to the pretrained model. Removing the feature pyramid completely significantly degrades the performance, with a 7.4% drop. Performance increases as we increase the number of convolution layers but saturates at 3. The best setup derived here is used by following experiments.

Variations on the text encoder and tokens. In Table 4.2, we freeze the CLIP image encoder, pair it with different text encoders, and finetune them. Using "un-

Table 4.3: Effect of number of text tokens. We show that using all text tokens (16 tokens for both TAL and AS and 32 tokens for MR) performs better than using a single token in video-text fusion on different tasks. Note that the image encoder is frozen.

# tokens	ANet TAL	ANet MR	COIN
All	**53.7**	**44.2**	**16.4**
One	53.3	42.6	15.7

paired" image-text encoders indeed diminishes the performance on all three tasks, especially for TAL and MR. For closed-vocabulary tasks, such as TAL, a text encoder is not strictly required. We hence compare our model to a version without the text encoder and try to make minimum changes to ensure a fair comparison. Without the text tokens the video-text fusion module becomes a temporal encoder (i.e. a transformer which operates on frame-level features, aggregating temporal information across them). To enable this ablation, we also modify the linear projections in Eqn. (4.1) and Eqn. (4.2) as follows:

$$\hat{\mathbf{Y}} = \mathbf{Z}\mathbf{W}_{cls} + \mathbf{b}_{cls} \tag{4.5}$$

$$\Delta\hat{\mathbf{T}} = \text{relu}\left(\mathbf{Z}\mathbf{W}_{reg} + \mathbf{b}_{reg}\right) \tag{4.6}$$

where $\mathbf{Z} \in \mathbb{R}^{N\times K}$ are the activiations after convolution layers, $\hat{\mathbf{Y}} \in \mathbb{R}^{N\times C}$ and $\Delta\hat{\mathbf{T}} \in \mathbb{R}^{N\times 2C}$ are the predicted class logits and start/end time displacements.

After removing the text encoder, the performance on ANet TAL drops from 54.7 mAP@ 0.5IoU to 46.5 (a relative decrease of 15%). In a second study, we also compare the performance of using a single text [CLS] token versus using all the text tokens from the text encoder on different tasks shown in Table 4.3. For close-vocabulary tasks, such as TAL and AS, all refers to 16 tokens to represent the class labels and for MR we increase the sequence length to 32 , i.e., captions contain more words than class labels. We demonstrate that using all tokens gives better performance on all tasks and such improvement is larger for tasks involved more complex language queries, such as MR.

Effect of video-text fusion module. We also compare our model with a late-fusion variant where the frame relevancy scores are computed as the dot product between the normalized $\mathbf{Z}$ and the class label text embeddings. This variant improves over the no-text variant to 49.8 on ANet TAL but still worse than our proposed mid-fusion model. We find that video-text fusion is essential for achieving good performance on TAL.

Table 4.4: Effect of freezing or finetuning image/text encoder on different tasks. The video-text fusion module and heads are always finetuned. For closed-vocabulary tasks, such as TAL and AS, finetune the image encoder is better (bottom two rows), however, for tasks involving more complex queries such as MR, finetuning the image encoder degrades performance (top two rows).

Image/Text encoders	ANet TAL	ANet MR	COIN
frozen/frozen	53.2	43.4	16.1
frozen/finetuned	53.3	**44.2**	16.4
finetuned/frozen	**54.7**	39.7	16.6
finetuned/finetuned	54.3	41.2	**16.9**

Finetuning strategies. Table 4.4 compares four different strategies for finetuning a Kinetics-pretrained model on downstream tasks by either freezing or finetuning each of the two encoders. In this study, we always finetune the video-text fusion layers and heads. We observe that it is more beneficial to finetune the image encoder for close vocabulary tasks, such as TAL and AS. However, for task involving more complex queries, such as MR, finetuning the image encoder actually degrades the performance. A similar phenomenon is also observed by [119], and may be due to overfitting.

4.3.3 Comparison with the state-of-the-art

In this section we compare to the state-of-the-art for all three tasks individually. Qualitative examples for each task are provided in Fig. 4.3.

Moment retrieval. For MR models we freeze the image encoder and finetune the rest of the network following the best strategy derived in Table 4.4. On ANet MR, our UnLoc-L model achieves a new state-of-the-art improving the previous best by 2.0% and 0.4% in recall @ 1 under $\mathrm{IoU} = 0.5$ and 0.7, respectively (Table 4.5). On Charades-STA, our UnLocL model improves upon the previous best [47] by 1.3% and 2.9% on the same two metrics. On ANet MR, UnLocL outperforms [47] by a larger margin, 6.8% and 7.1%. On QVHighlights, UnLoc- L improves upon the previous best [122] by 3.7% and 1.7%. Most previous work is built upon pre-extracted convolutional features, such as I3D [123], P3D [124], C3D [125], R(2+1)D [126], VGG [127], SlowFast [128], etc., and our work is most comparable to [48], which also employs CLIP features (in addition to SlowFast [128] features). Our UnLoc-L model scores 5.1% and 4.4% higher than [48] on Charades-STA in recall@ 1 under IoU=0.5 and 0.7 . To the best of our knowledge,

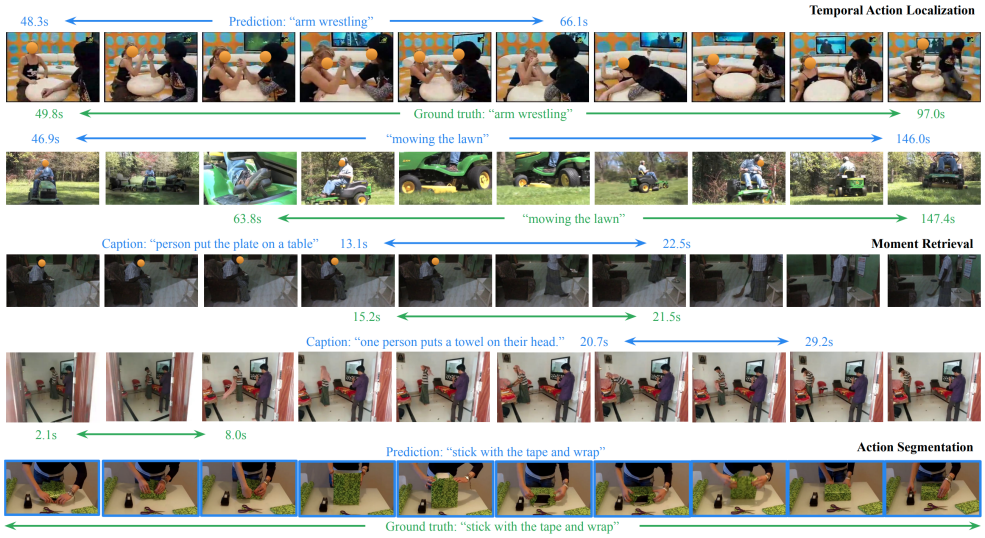


Figure 4.3: Qualitative Results We show results on ActivityNet, Charades and COIN, for Temporal Action Localization, Moment Retrieval and Action Segmentation respectively. Predictions are shown in blue, while the ground truth is in green (best viewed in colour). For action segmentation, the ground truth covers the entire clip. Note how our model is able to predict accurate boundaries, in some cases better refined than the ground truth (top row, the arm wrestling action has stopped, however the ground truth boundary extends for a while after). For the second example for Moment Retrieval (4th row from top), we show a failure case, where our model detects the moment where the towel is 'put down', and not 'on their head' as perhaps the latter is a rarer occurrence in the training data.

Table 4.5: Comparison with the state-of-the-art for Moment Retrieval. We show results on Charades-STA (test split), ANet MR (val_2 split), and QVHighlights (val split) datasets.

	Method	Vision Enc.	R@1		R@5	
			IoU = 0.5	IoU = 0.7	IoU = 0.5	IoU = 0.7
Charades-STA	CTRL [40]	C3D	23.6	8.9	58.9	29.5
	2D TAN [46]	VGG	39.7	23.3	80.3	51.3
	VSLNet [120]	I3D	47.3	30.2	-	-
	UMT [49]	VGG	49.4	26.2	**89.4**	55.0
	IVG-DCL [121]	C3D	50.2	32.9	-	-
	M-DETR [48]	CLIP	55.7	34.2	-	-
	LGI [47]	I3D	59.5	35.5	-	-
	UnLoc-B	CLIP	58.1	35.4	87.4	59.1
	UnLoc-L	CLIP	**60.8**	**38.4**	88.2	**61.1**
ANet MR	LGI [47]	C3D	41.5	23.1	-	-
	VSLNet [120]	I3D	43.2	26.2	-	-
	2D TAN [46]	C3D	44.5	26.5	77.1	62.0
	DRN [45]	C3D	45.5	24.4	78.0	50.3
	VLG [98]	C3D	46.3	29.8	77.2	**63.3**
	UnLoc-B	CLIP	48.0	29.7	**81.5**	61.4
	UnLoc-L	CLIP	**48.3**	**30.2**	79.2	61.3
QVHighlights	M-DETR [48]	SF+CLIP	53.9	34.8	-	-
	UMT [49]	SF+ CLIP	60.3	44.3	-	-
	QD-DETR [122]	SF+CLIP	62.4	45.0	-	-
	UnLoc-B	CLIP	64.5	**48.8**	-	-
	UnLoc-L	CLIP	**66.1**	46.7	-	-

Table 4.6: Comparison with the state-of-the-art on ANet TAL. We show results for finetuning, and both the zero-shot (open-set) protocols introduced by [37]. Our method outperforms all previous work across all settings, achieving strong gains particularly in the zero-shot settings.

Setting	Method	Vision Encoder	mAP@0.5IoU
Finetuned	A2Net [129]	I3D	43.6
	TSP [130]	R(2+1)D	51.3
	GTAN [131]	P3D	52.6
	VSGN [132]	I3D	53.3
	TadTR [133]	R(2+1)D	53.6
	PBRNet [134]	I3D	54.0
	TCANet [135]	SlowFast	54.3
	ActionFormer [36]	R(2+1)D	54.7
	ContextLoc [136]	I3D	56.0
	EffPrompt [37]	CLIP	44.0
	STALE [38]	CLIP	54.3
	STALE [38]	I3D	56.5
	UnLoc-B (1st prompt)	CLIP	54.6
	UnLoc-L (1st prompt)	CLIP	58.8
	UnLoc-L (prompt ensembling)	CLIP	**59.3**
Zero-shot 50% Seen 50% Unseen	EffPrompt [37]	CLIP	32.0
	STALE [38]	CLIP	32.1
	UnLoc-B (1st prompt)	CLIP	36.9
	UnLoc-L (1st prompt)	CLIP	43.2
	UnLoc-L (prompt ensembling)	CLIP	**43.7**
Zero-shot 75% Seen 25% Unseen	EffPrompt [37]	CLIP	37.6
	STALE [38]	CLIP	38.2
	UnLoc-B (1st prompt)	CLIP	40.2
	UnLoc-L (1st prompt)	CLIP	47.4
	UnLoc-L (prompt ensembling)	CLIP	**48.8**

we are the first work employing pure transformer features that achieves state-of-the-art results on moment retrieval, which has largely been dominated by CNN-based features.

Temporal localization. Table 4.6 shows results on ANet TAL under two settings (finetuned and zero-shot). In the finetuned setting, we freeze the text encoder and finetune the rest of the network following the best strategy derived from Table 4. For UnLoc-L we increase the sampled frames to 160 and use a 5-L Feature Pyramid. As shown in Table 4.6, most high-performance methods are built on top of 3D convolutional features. There are two previous attempts to replace the CNN vision encoder by a Transformer encoder. EffPrompt [37], built on top of frozen CLIP features, scored significantly lower than recent CNN-based models

Table 4.7: Comparison with the state-of-the-art on COIN for Action Segmentation. We report results using both frame accuracy (as is standard practice) and mAP, which we believe is a better metric given that a large proportion (58.9%) of the dataset is labelled as a single class (background).

Method	Frame accuracy	mAP
Baseline: predict all background	58.9	0.0
ActBERT [137]	57.0	-
MIL-NCE [54]	61.0	-
TACo [138]	68.4	-
VLM [139]	68.4	-
VideoCLIP [55]	68.7	-
UniVL [56]	70.0	-
UnLoc-B (1st prompt)	68.0	36.2
UnLoc-L (1st prompt)	72.6	47.0
UnLoc-L (prompt ensembling)	**72.8**	47.7

and STALE [38], which is also built upon CLIP features, achieved competitive results with the best CNN methods but is 2.2 worse than the same model trained on two-stream I3D features. To the best of our knowledge, we are the first work that achieved state-of-the-art results using only Transformer features. Our UnLoc-L model improved previous best results in terms of mAP@ $0.5\mathrm{IoU}$ by 2.3 and with prompt ensembling this margin is increased to 2.8 .

For both splits in the zero-shot (open-set) protocols proposed by [37, 38], UnLoc-B and L outperform previous best by a significant margin. Specifically, UnLoc-L advances previous state-of-the-art by 11.6 , a relative 36.1% improvement on the $50/50$ split and by 10.6 , a relative 27.7% on the $75/25$ split.

Action segmentation. Table 4.7 compares our model with previous work and UnLoc-L outperform previous state-ofthe-art by 2.8% in frame accuracy. Besides architectural differences, we note that previous works [54, 55, 56] pretrain their models on HowTo100M [57], which consists of around $100\mathrm{M}$ aligned ASR and video clip pairs, and is also in a similar domain to COIN (instructional web videos). Our models on the other hand, are initialized from CLIP checkpoints, which are trained on cleaner web image-text pairs from multiple domains and finetuned on Kinetics, 10s clips of human activity videos.

4.4 Conclusions

We propose a new model for video localization tasks, called UnLoc. UnLoc consists of a two-tower CLIP model, the output features of which are fed into a video-text fusion module and feature pyramid. Unlike previous works, we achieve state-of-the-art results on 3 different benchmarks (moment retrieval, temporal action localization and action segmentation) with a single approach, without the need for action proposals or pretrained video features.

Future work will investigate cotraining on the three localization tasks, pretraining on large, weakly labelled datasets, exploring highlight detection as an additional downstream task, and adapting our model to other modalities such as audio for sound localization [140].

CHAPTER 5

HIFI TUNER: HIGH-FIDELITY SUBJECT-DRIVEN FINE-TUNING FOR DIFFUSION MODELS

5.1 Introduction

Diffusion models [141, 142] have demonstrated a remarkable success in producing realistic and diverse images. The advent of large-scale text-to-image diffusion models [62, 61, 143], leveraging expansive web-scale training datasets [144, 145], has enabled the generation of high-quality images that align closely with textual guidance. Despite this achievement, the training data remains inherently limited in its coverage of all possible subjects. Consequently, it becomes infeasible for diffusion models to accurately generate images of specific, unseen subjects based solely on textual descriptions. As a result, personalized generation has emerged as a pivotal research problem. This approach seeks to fine-tune the model with minimal additional costs, aiming to generate images of user-specified subjects that seamlessly align with the provided text descriptions.

We identify three drawbacks of existing popular methods for subject-driven fine-tuning [68, 146, 69, 147]. Firstly, a notable imbalance exists between sample quality and parameter efficiency in the fine-tuning process. For example, Textual Inversion optimizes only a few parameters in the text embedding space, resulting in poor sample fidelity. Conversely, DreamBooth achieves commendable sample fidelity but at the cost of optimizing a substantial number of parameters. Ideally, there should be a parameter-efficient method that facilitates the generation of images with satisfactory sample fidelity while remaining lightweight for improved portability. Secondly, achieving a equilibrium between sample fidelity and the flexibility to render objects in diverse scenes poses a significant challenge. Typically, as fine-tuning iterations increase, the sample fidelity improves, but the flexibility of the scene coverage diminishes. Thirdly, current methods struggle to accurately preserve the appearance of the input object. Due to the extraction of subject representations from limited data, these representations offer weak con-

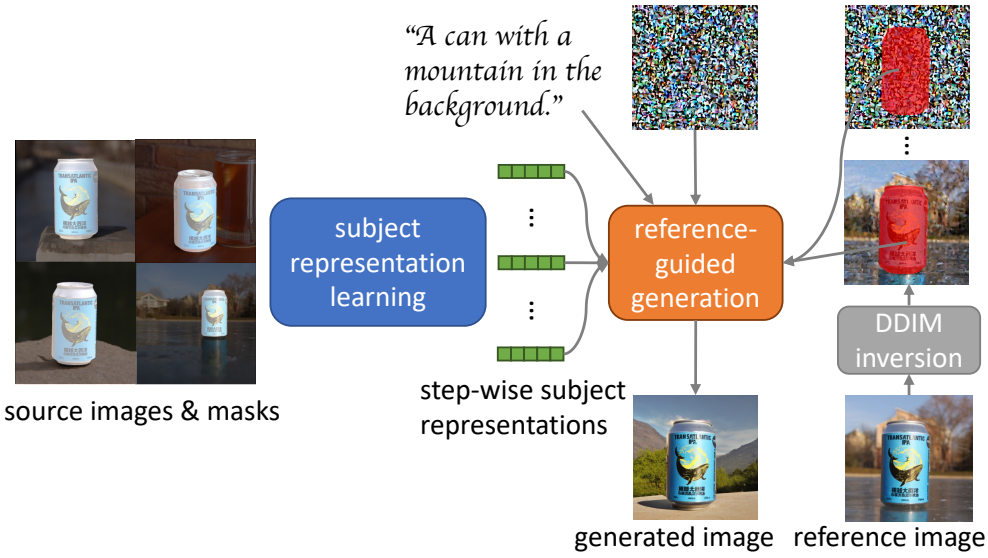


Figure 5.1: Illustration of HiFi Tuner. We first learn the step-wise subject representations with subject source images and masks. Then, we select and transform the reference image, and use DDIM inversion to obtain its noise latent trajectory. Finally, we generate an image controlled by the prompt, the step-wise subject representations and the reference subject guidance.

straints to the diffusion model. Consequently, unwanted variations and artifacts may appear in the generated subject.

In this study, we introduce a novel framework named HiFi Tuner for subject fine-tuning that prioritizes improving the sample fidelity while preserving the scene coverage. We first introduce a step-wise subject representation learning strategy that distinguishes the functions of subject representations at different denoising time steps. This strategy incorporates a mask guidance to reduce the influence of the image background for subject representations and a novel parameter regularization method to sustain the model's scene coverage capability. Then, we propose a reference-guided generation method that leverages pivotal inversion of a reference image. By integrating guiding information into the step-wise denoising process, we effectively mitigate the unwanted variations and artifacts in the generated subjects. Furthermore, our framework demonstrates versatility by extending its application to a novel image editing task: substituting the subject in an image with a user-specified subject through textual manipulations.

We summarize the contributions of our work as follows. Firstly, we propose a step-wise subject representation learning strategy that significantly helps the diffusion model generate samples with improved sample fidelity. Secondly, we

introduce a novel reference-guided generation process that successfully addresses unwanted subject variations and artifacts in the generated images. Thirdly, we extend the application of our methodology to a new subject-driven image editing task, showcasing its versatility and applicability in diverse scenarios. Finally, we demonstrate the generic nature of HiFi Tuner by showcasing its effectiveness in enhancing the performance of both the Textual Inversion and the DreamBooth, which results in a new state of the art in subject-driven fine-tuning for diffusion models.

5.2 Methods

In this section, we elaborate HiFi Tuner in details. We show the framework of HiFi Tuner in Fig. 5.2. In section 5.2.1, we present some necessary backgrounds for our work. In section 5.2.2, we introduce the step-wise subject representation learning strategy that helps preserving the subject identity. In section 5.2.3, we introduce the reference-guided generation technique, which merits the image inversion process to better preserve subject details. In section 5.2.4, we introduce an extension of our work on a novel image editing application – personalized subject replacement with only textual prompt edition.

5.2.1 Backgrounds

Stable diffusion [61] is a widely adopted framework in the realm of text-to-image diffusion models. Unlike other methods [62, 143], Stable diffusion is a latent diffusion model, where the diffusion model is trained within the latent space of a Variational Autoencoder (VAE). To accomplish text-to-image generation, a text prompt undergoes encoding into textual embeddings c using a CLIP text encoder[91]. Subsequently, a random Gaussian noise latent x_T is initialized. The process then recursively denoises noisy latent x_t through a noise predictor network ϵ_θ with the conditioning of c. Finally, the VAE decoder is employed to project the denoised latent x_0 onto an image. During the sampling process, a commonly applied mechanism involves classifier-free guidance [149] to enhance sample quality. Additionally, deterministic samplers, such as DDIM [150], are employed to improve sampling efficiency. The denoising process can be expressed

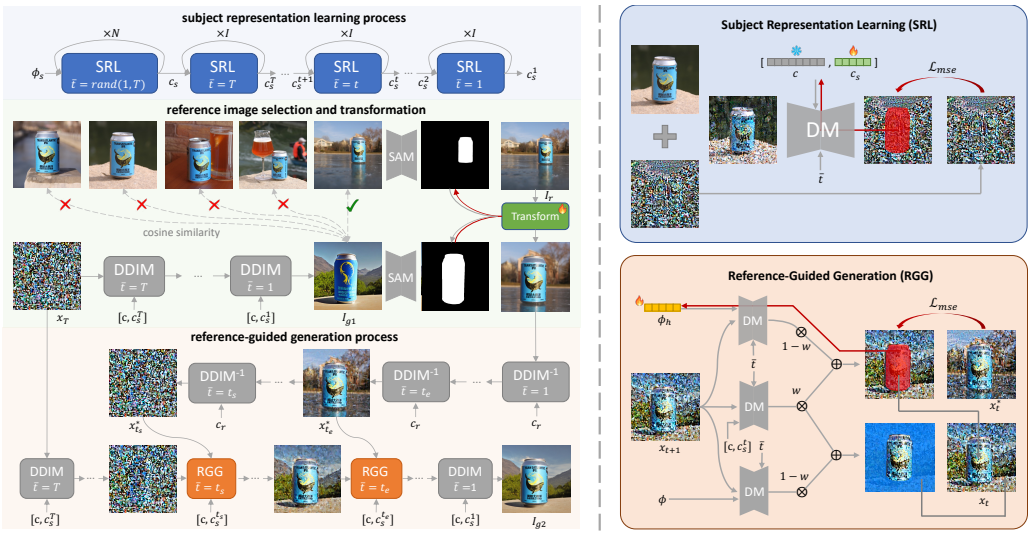


Figure 5.2: The framework of HiFi Tuner. The grey arrows stand for the data flow direction. The red arrows stand for the gradient back propagation direction. SAM stands for the Segment Anything [148] model. DM stands for the Stable Diffusion [61] model. $DDIM$ and $DDIM^{-1}$ stands for the DDIM denoising step and inversion step respectively.

as

$$\begin{aligned} x_{t-1} &= F^{(t)}(x_t, c, \phi) \\ &= \beta_t x_t - \gamma_t(w\epsilon_\theta(x_t, c) + (1-w)\epsilon_\theta(x_t, \phi)). \end{aligned} \tag{5.1}$$

where β_t and γ_t are time-dependent constants; w is the classifier-free guidance weight; ϕ is the CLIP embedding for a null string.

Textual inversion [69]. As a pioneer work in personalized generation, Textual Inversion introduced the novel concept that a singular learnable textual token is adequate to represent a subject for the personalization. Specifically, the method keeps all the parameters of the diffusion model frozen, exclusively training a word embedding vector c_s using the diffusion objective:

$$\mathfrak{L}_s(c_s) = \min_{c_s} \|\epsilon_\theta(x_t, [c, c_s]) - \epsilon\|_2^2, \tag{5.2}$$

where $[c, c_s]$ represents replacing the object-related word embedding in the embedding sequence of the training caption (e.g. "a photo of A") with the learnable embedding c_s. After c_s is optimized, this work applies $F^{(t)}(x_t, [c, c_s], \phi)$ for generating personalized images from prompts.

Null-text inversion [74] method introduces an inversion-based approach to im-

age editing, entailing the initial inversion of an image input to the latent space, followed by denoising with a user-provided prompt. This method comprises two crucial processes: a pivotal inversion process and a null-text optimization process. The pivotal inversion involves the reversal of the latent representation of an input image, denoted as x_0, back to a noise latent representation, x_T, achieved through the application of reverse DDIM. This process can be formulated as reparameterizing Eqn. (5.1) with $w = 1$:

$$x_{t+1} = F^{-1(t)}(x_t, c) = \overline{\beta_t} x_t + \overline{\gamma_t} \epsilon_\theta(x_t, c) \tag{5.3}$$

We denote the latent trajectory attained from the pivotal inversion as $[x_0^*, ..., x_T^*]$. However, naively applying Eqn. (5.1) for x_T^* will not restore x_0^*, because $\epsilon_\theta(x_t, c) \neq \epsilon_\theta(x_{t-1}^*, c)$. To recover the original image, Null-text inversion trains a null-text embedding ϕ_t for each timestep t force the the denoising trajectory to stay close to the forward trajectory $[x_0^*, ..., x_T^*]$. The learning objective is

$$\mathfrak{L}_h^{(t)}(\phi_t) = \min_{\phi_t} \|x_{t-1}^* - F^{(t)}(x_t, c, \phi_t)\|_2^2. \tag{5.4}$$

After training, image editing techniques such as the prompt-to-prompt [73] can be applied with the learned null-text embeddings $\{\phi_t^*\}$ to allow manipulations of the input image.

5.2.2 Learning step-wise subject representations

We observe that the learned textual embedding, c_s, plays distinct roles across various denoising time steps. In early time steps where t is large, the primary focus is on generating high-level image structures, while at smaller values of t, the denoising process shifts its emphasis toward refining finer details. Our analysis of c_s across time steps, presented in Fig. 5.3, underscores these variations. To address this issue, we first propose a better loss function combining a mask guidance and a parameter regularization; then, we come up with a method to distinguish subject representations per time step.

The subject representations, c_s, in textual inversion is susceptible to be influenced by the backgrounds of the training images. This influence often imposes constraints on the style and scene of generated samples and makes the identity preservation more difficult. To address this issue, we propose to use subject masks

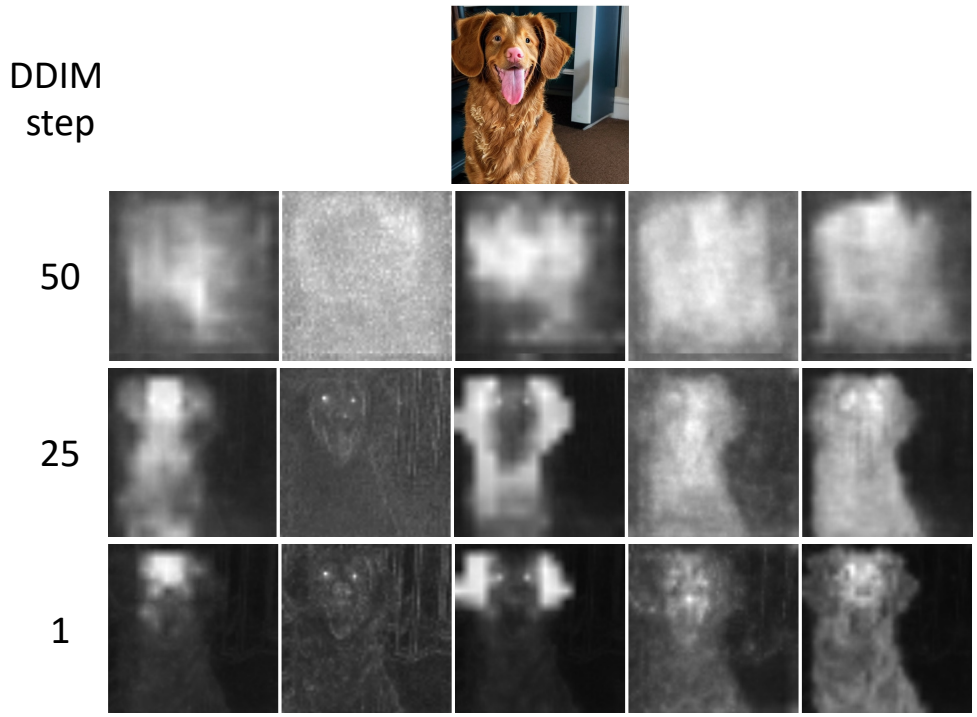


Figure 5.3: Step-wise function analysis of c_s. We generate an image from a noise latent with DDIM and an optimized c_s representing a subject dog. The text prompt is "A sitting dog". The top image is the result generated image. We follow [73] to obtain the attention maps with respect to the 5 token embeddings of c_s as shown in the below images. The numbers to the left refer to the corresponding DDIM denoising steps. In time step 50, the 5 token embeddings of c_s are attended homogeneously across the latent vectors. In time step 1, these token embeddings are attended mostly by the subject detailed regions such as the forehead, the eyes, the ears, etc.

to confine the loss during the learning process of c_s. This approach ensures that the training of c_s predominantly focuses on subject regions. We use Segment Anything (SAM) [148] to obtain binary masks of the subjects in the source images. Then, we introduce a regularization term to achieve the equilibrium between the identity preservation and the ability of generating diverse scenes. The Eqn. (5.2) is thus updated to a new loss:

$$\mathfrak{L}_s(c_s) = \min_{c_s} \|M \odot (\epsilon_\theta(x_t, [c, c_s]) - \epsilon)\|_2^2 + w_s \|c_s - \phi_s\|_2^2. \tag{5.5}$$

where $\odot$ stands for element-wise product, M stands for a binary mask of the subject. $c_s \in \mathbb{R}^{n \times d}$ where n is the number of tokens and d is the embedding dimension, and w_s is a regularization hyper-parameter. We define ϕ_s as the last n embeddings of ϕ. Substituting the last n embeddings in c with c_s forms $[c, c_s]$. It is noteworthy that $[c, c_s] = c$ if c_s is not optimized, given that ϕ constitutes the padding part of the embedding. This regularization serves two primary purposes.

Algorithm 3: Optimization algorithm for c_s^t. T is DDIM time steps. I is the optimization steps per DDIM time step. X_0 is the set of encoded latents of the source images. $N_s(\cdot)$ is the DDIM noise scheduler. $\mathfrak{L}_s(\cdot)$ refers to the loss function in Eqn. (5.5).

Result: C_s
$C_s = \{\}$, $c_s^{T+1} = c_s$
for $t = [T, ..., 1]$ **do**
  $c_s^t = c_s^{t+1}$
  for $i = [1, ..., I]$ **do**
    $\epsilon \sim \mathcal{N}(0, 1)$, $x_0 \in X_0$, $x_t = N_s(x_0, \epsilon, t)$
    $c_s^t = c_s^t - \eta \nabla_{c_s^t} \mathfrak{L}_s(c_s^t)$
  $C_s = C_s \cup \{c_s^t\}$

Firstly, the stable diffusion model is trained with a 10% caption drop, simplifying the conditioning to ϕ and facilitating classifier-free guidance [149]. Consequently, ϕ is adept at guiding the diffusion model to generate a diverse array of scenes, making it an ideal anchor point for the learned embedding. Secondly, due to the limited data used for training the embedding, unconstrained parameters may lead to overfitting with erratic scales. This overfitting poses a risk of generating severely out-of-distribution textual embeddings.

To distinguish the subject representations at different denoising stages, we introduce time-dependent embeddings, c_s^t, at each time step instead of a single c_s to represent the subject. This leads to a set of embeddings, $[c_s^1, ..., c_s^T]$, working collectively to generate images. To ensure smooth transitions between time-dependent embeddings, we initially train a single c_s across all time steps. Subsequently, we recursively optimize c_s^t following DDIM time steps, as illustrated in Algorithm 3. This approach ensures that c_s^t is proximate to c_s^{t+1} by initializing it with c_s^{t+1} and optimizing it for a few steps. After training, we apply

$$x_{t-1} = F^{(t)}(x_t, [c, c_s^t], \phi) \tag{5.6}$$

with the optimized $[c_s^1, ..., c_s^T]$ to generate images.

5.2.3 Reference-guided generation

Shown in Figure 5.2, we perform our reference-guided generation in three steps. First, we determine the initial latent x_T and follow the DDIM denoising process to generate an image. Thus, we can determine the subject regions of $\{x_t\}$ re-

quiring guiding information and the corresponding reference image. Second, we transform the reference image and inverse the latent of the transformed image to obtain a reference latent trajectory, $[x_0^*, ..., x_T^*]$. Third, we start a new denoising process from x_T and apply the guiding information from $[x_0^*, ..., x_T^*]$ to the guided regions of $\{x_t\}$. Thereby, we get a reference-guided generated image.

Guided regions and reference image. First, we determine the subject regions of x_t that need the guiding information. Notice that $x_t \in \mathbb{R}^{H\times W\times C}$, where H, W and C are the height, width and channels of the latent x_t respectively. Following the instance segmentation methods [77, 151], we aim to find a subject binary mask M_g to determine the subset $x_t^s \in \mathbb{R}^{m\times C}$ corresponding to the subject regions. Because DDIM [150] is a deterministic denoising process as shown in Eqn. (5.1), once x_T, c and ϕ are determined, the image to be generated is already determined. Therefore, we random initialize x_T with Gaussian noise; then, we follow Eqn. (5.6) and apply the decoder of the stable diffusion model to obtain a generated image, I_{g1}; by applying Grounding SAM [152, 148] with the subject name to I_{g1} and resizing the result to $H \times W$, we obtain the subject binary mask M_g. Second, we determine the reference image by choosing the source image with the closest subject appearance to the subject in I_{g1}, since the reference-guided generation should modify $\{x_t\}$ as small as possible to preserve the image structure. As pointed out by DreamBooth [68], DINO [153] score is a better metric than CLIP-I [91] score in measuring the subject similarity between two images. Hence, we use ViT-S/16 DINO model [153] to extract the embedding of I_{g1} and all source images. We choose the source image whose DINO embedding has the highest cosine similarity to the DINO embedding of I_{g1} as the reference image, I_r. We use Grounding SAM [152, 148] to obtain the subject binary mask M_r of I_r.

Reference image transformation and inversion. First, we discuss the transformation of I_r. Because the subject in I_{g1} and the subject in I_r are spatially correlated with each other, we need to transform I_r to let the subject better align with the subject in I_{g1}. As the generated subject is prone to have large appearance variations, it is noneffective to use image registration algorithms, e.g. RANSAC [154], based on local feature alignment. We propose to optimize a transformation matrix

$$T_\theta = \begin{bmatrix} \theta_1 & 0 & 0 \\ 0 & \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_2) & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos(\theta_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \theta_3 \\ 0 & 1 & \theta_4 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.7}$$

composed of scaling, rotation, and translation such that $T_\theta(M_r)$ best aligns with

Algorithm 4: Reference-guided generation algorithm. J is the number of optimization steps for ϕ_h per denoising step. $\mathfrak{L}_h^{(t)}(\cdot)$ refers to the loss function in Eqn. (5.9).

Result: x_0
Inputs: t_s, t_e, x_T, M_r^*, c, ϕ, $[c_s^1, ..., c_s^T]$, $[x_0^*, ..., x_T^*]$
for $t = [T, ..., 1]$ **do**
  if $t == t_s$ **then**
    $\phi_h = \phi$
    $x_t[M_r^*] = x_t^*[M_r^*]$
  $x_{t-1} = F^{(t)}(x_t, [c, c_s^t], \phi)$
  if $t \leqslant t_s$ ***and*** $t \geqslant t_e$ **then**
    for $j = [1, ..., J]$ **do**
      $\phi_h = \phi_h - \eta \nabla_{\phi_h} \mathfrak{L}_h^{(t)}(\phi_h)$
    $x_{t-1}[M_r^*] = F^{(t)}(x_t, [c, c_s^t], \phi_h)[M_r^*]$

M_g. Here, $\{\theta_i\}$ are learnable parameters, and $T_\theta(\cdot)$ is the function of applying the transformation to an image. T_θ can be optimized with

$$\mathfrak{L}_t = \min_\theta \|T_\theta(M_r) - M_g\|_1^1. \tag{5.8}$$

Please refer to Appendix B.1 for a specific algorithm optimizing T_θ. We denote the optimized T_θ as T_θ^* and the result of $T_\theta^*(M_r)$ as M_r^*. Thereafter, we can transform I_r with $T_\theta^*(I_r)$ to align the subject with the subject in I_{g1}. Notice that the subject in $T_\theta^*(I_r)$ usually does not perfectly align with the subject in I_{g1}. A rough spatial location for placing the reference subject should suffice for the reference guiding purpose in our case. Second, we discuss the inversion of $T_\theta^*(I_r)$. We use BLIP-2 model [155] to caption I_r and use a CLIP text encoder to encode the caption to c_r. Then, we encode $T_\theta^*(I_r)$ into x_0^* with a Stable Diffusion image encoder. Finally, we recursively apply Eqn. (5.3) to obtain the reference latent trajectory, $[x_0^*, ..., x_T^*]$.

Generation process. There are two problems with the reference-guided generation: 1) the image structure needs to be preserved; 2) the subject generated needs to conform with the context of the image. We reuse x_T in step 1 as the initial latent. If we follow Eqn. (5.6) for the denoising process, we will obtain I_{g1}. We aim to add guiding information to the denoising process and obtain a new image I_{g2} such that the subject in I_{g2} has better fidelity and the image structure is similar to I_{g1}. Please refer to Algorithm 4 for the specific reference-guided generation process. As discussed in Section 5.2.2, the stable diffusion model focuses on the image structure formation at early denoising steps and the detail polishing at later

steps. If we incur the guiding information in early steps, I_{g2} is subject to have structural change such that M_r^* cannot accurately indicate the subject regions. It is harmful to enforce the guiding information at later steps either, because the denoising at this stage gathers useful information mostly from the current latent. Therefore, we start and end the guiding process at middle time steps t_s and t_e respectively. At time step t_s, we substitute the latent variables corresponding to the subject region in x_t with those in x_t^*. We do this for three reasons: 1) the substitution enables the denoising process to assimilate the subject to be generated to the reference subject; 2) the latent variables at time step t_s are close to the noise space so that they are largely influenced by the textual guidance as well; 3) the substitution does not drastically change the image structure because latent variables have small global effect at middle denoising steps. We modify Eqn. (5.4) to Eqn. (5.9) for guiding the subject generation.

$$\mathfrak{L}_h^{(t)}(\phi_h) = \min_{\phi_h} \|x_{t-1}^*[M_r^*] - F^{(t)}(x_t, [c, c_s^t], \phi_h)[M_r^*]\|_2^2 \tag{5.9}$$

Here, $x_t[M]$ refers to latent variables in x_t indicated by the mask M. Because ϕ_h is optimized with a few steps per denoising time step, the latent variables corresponding to the subject regions change mildly within the denoising time step. Therefore, at the next denoising time step, the stable diffusion model can adapt the latent variables corresponding to non-subject regions to conform with the change of the latent variables corresponding to the subject regions. Furthermore, we can adjust the optimization steps for ϕ_h to determine the weight of the reference guidance. More reference guidance will lead to a higher resemblance to the reference subject, while less reference guidance will result in more variations for the generated subject.

5.2.4 Personalized subject replacement

We aim to use the learned subject textual representations to replace the subject in an image with the user-specified subject. Although there are methods [156, 157, 158, 159] inpainting the image area with a user-specified subject, our method has two advantages over them. First, we do not specify the inpainting area of the image; instead, our method utilize the correlation between the textual embeddings and the latent variables to identify the subject area. Second, our method can generate a subject with various pose and appearance, such that the

Algorithm 5: Personalized subject replacement algorithm. $F^{-1(t)}$ refers to Eqn. (5.3). K is the optimization steps for null-text optimization. $\mathfrak{L}_h^{(t)}(\cdot)$ refers to Eqn. (5.4)

Result: x_0^g
Inputs: x_0^r, c^r, c^g, $[c_s^1, ..., c_s^T]$
$x_0^{r*} = x_0^r$
for $t = [0, ..., T-1]$ **do**
  $x_{t+1}^{r\,*} = F^{-1(t)}(x_t^{r*}, c^r)$
$x_T^r = x_T^{r\,*}$, $\phi_T = \phi$
for $t = [T, ..., 1]$ **do**
  for $k = [1, ..., K]$ **do**
    $\phi_t = \phi_t - \eta \nabla_{\phi_t} \mathfrak{L}_h^{(t)}(\phi_t)$
  $x_{t-1}^r, a_t^{r*} = A^{(t)}(x_t^r, c^r, \phi_t)$
  $\phi_{t-1} = \phi_t^* = \phi_t$
$x_T^g = x_T^{r\,*}$
for $t = [T, ..., 1]$ **do**
  $x_{t-1}^g = \tilde{F}_{[c_s^t, w_g]}^{(t)}(x_t^g, [c^g, c_s^t], \phi_t^*, a_t^{r*})$

added subject better conforms to the image context.

We first follow the fine-tuning method in Section 5.2.2 to obtain the step-wise subject representations $[c_s^1, ..., c_s^T]$. We encode the original image I_r to x_0^r with the Stable Diffusion image encoder; then, we use BLIP-2 model [155] to caption I_r and encode the caption into c^r with the Stable Diffusion language encoder. We identify the original subject word embedding in c^r and substitute that with the new subject word embedding w_g to attain a c^g (e.g. 'cat' $\rightarrow$ 'dog' in the sentence 'a photo of siting cat'). Then, we follow Algorithm 5 to generate the image with the subject replaced. Referring to the prompt-to-prompt paper [73], we store the step-wise cross attention weights with regard to the word embeddings in c^r to a_t^{r*}. $A^{(t)}(\cdot, \cdot, \cdot)$ performs the same operations as $F^{(t)}(\cdot, \cdot, \cdot)$ in Eqn. (5.1) but returns x_{t-1} and a_t^{r*}. We also modify $F^{(t)}(\cdot, \cdot, \cdot)$ to $\tilde{F}_{[c_s^t, w_g]}^{(t)}(\cdot, \cdot, \cdot, a_t^{r*})$ such that all token embeddings use fixed cross attention weights a_t^{r*} except that $[c_s^t, w_g]$ use the cross attention weights of the new denoising process.

5.3 Experiments

Dataset. We use the DreamBooth [68] dataset for evaluation. It contains 30 subjects: 21 of them are rigid objects and 9 of them are live animals subject to large appearance variations. The dataset provides 25 prompt templates for generating

Figure 5.4: Qualitative comparison. We implement our fine-tuning method based on both Textual Inversion (TI) and DreamBooth (DB). A visible improvement is made by comparing the images in the third column with those in the second column and comparing the images in the fifth column and those in the forth column.

images. Following DreamBooth, we fine-tune our framework for each subject and generate 4 images for each prompt template, totaling 3,000 images.

Settings. We adopt the pretrained Stable Diffusion [61] version 1.4 as the text-to-image framework. We use DDIM with 50 steps for the generation process. For HiFi Tuner based on Textual Inversion, we implement both the learning of subject textual embeddings described in Section 5.2.2 and the reference-guided generation described in Section 5.2.3. We use 5 tokens for c_s and adopts an ADAM [160] optimizer with a learning rate $5e^{-3}$ to optimize it. We first optimize c_s for 1000 steps and then recursively optimize c_s^t for 10 steps per denoising step. We set $t_s = 40$ and $t_e = 10$ and use an ADAM [160] optimizer with a learning rate $1e^{-2}$ to optimize ϕ_h. We optimize ϕ_h for 10 steps per DDIM denoising step. For

Table 5.1: Quantitative comparison.

Method	DINO ↑	CLIP-I ↑	CLIP-T ↑
Real images	0.774	0.885	N/A
Stable Diffusion	0.393	0.706	0.337
Textual Inversion [69]	0.569	0.780	0.255
Ours (Textual Inversion)	0.665	0.807	0.291
DreamBooth [68]	0.668	0.803	0.305
Ours (DreamBooth)	**0.680**	**0.809**	**0.317**

Table 5.2: State-of-the-art comparison. *Anydoor is an subject inpainting method and thus does not measure vision-language alignment with CLIP-T score.

Method	DINO ↑	CLIP-I ↑	CLIP-T ↑
Textual Inversion [69]	0.569	0.780	0.255
ViCo [161]	0.631	0.809	0.229
BLIP-diffusion [162]	0.670	0.805	0.302
DreamBooth [68]	0.668	0.803	0.305
AnyDoor* [163]	0.678	**0.821**	-
Ours (Textual Inversion)	0.665	0.807	0.291
Ours (DreamBooth)	**0.680**	0.809	**0.317**

HiFi Tuner based on DreamBooth, we follow the original subject representation learning process and implement the reference-guided generation described in Section 5.2.3. We use the same optimization schedule to optimize ϕ_h as mentioned above. For the reference-guided generation, we only apply HiFi Tuner to the 21 rigid objects, because their appearances vary little and have strong need for the detail preservation.

Evaluation metrics. Following DreamBooth [68], we use DINO score and CLIP-I score to measure the subject fidelity and use CLIP-T score the measure the prompt fidelity. CLIP-I score is the average pairwise cosine similarity between CLIP [91] embeddings of generated images and real images, while DINO score calculates the same cosine similarity but uses DINO [153] embeddings instead of CLIP embeddings. As pointed out in the DreamBooth paper [68], DINO score is a better means than CLIP-I score in measuring the subject detail preservation. CLIP-T score is the average cosine similarity between CLIP [91] embeddings of the pairwise prompts and generated images.

Qualitative comparison. Fig. 5.4 shows the qualitative comparison between HiFi Tuner and other fine-tuning frameworks. HiFi Tuner possesses three ad-

Table 5.3: Ablation study.

Method	DINO ↑	CLIP-I ↑	CLIP-T ↑
Baseline (Textual Inversion)	0.567	0.786	0.293
+ mask and regularization	0.612	0.789	0.294
+ step-wise representations	0.626	0.790	0.292
+ reference guidance	0.665	0.807	0.291
Baseline (DreamBooth)	0.662	0.803	0.315
+ reference guidance	0.680	0.809	0.317

vantages compared to other methods. First, HiFi Tuner is able to diminish the unwanted style change for the generated subjects. As shown in Fig. 5.4 (a) & (b), DreamBooth blends sun flowers with the backpack, and both DreamBooth and Textual Inversion generate backpacks with incorrect colors; HiFi Tuner maintains the styles of the two backpacks. Second, HiFi Tuner can better preserve details of the subjects. In Fig. 5.4 (c), Textual Inversion cannot generate the whale on the can while DreamBooth generate the yellow part above the whale differently compared to the original image; In Fig. 5.4 (d), DreamBooth generates a candle with a white candle wick but the candle wick is brown in the original image. Our method outperforms Textual Inversion and DreamBooth in preserving these details. Third, HiFi Tuner can better preserve the structure of the subjects. In Fig. 5.4 (e) & (f), the toy car and the toy robot both have complex structures to preserve, and Textual Inversion and DreamBooth generate subjects with apparent structural differences. HiFi Tuner makes improvements on the model's structural preservation capability.

Quantitative comparison. We show the quantitative improvements HiFi Tuner makes in Table 5.1. HiFi Tuner improves Textual Inversion for 9.6 points in DINO score and 3.6 points in CLIP-T score, and improves DreamBooth for 1.2 points in DINO score and 1.2 points in CLIP-T score. We also compare our methods to the state-of-the-art methods shown in Fig. 5.2.

Ablation studies. We present the quantitative improvements of adding our proposed techniques in Table 5.3. We observe that fine-tuning either DreamBooth or Textual Inversion with more steps leads to a worse prompt fidelity. Therefore, we fine-tune the networks with fewer steps than the original implementations, which results in higher CLIP-T scores but lower DINO scores for the baselines. Thereafter, we can use our techniques to improve the subject fidelity so that both DINO scores and CLIP-T scores can surpass the original implementations. For HiFi Tuner based on Textual Inversion, we fine-tune the textual embeddings with

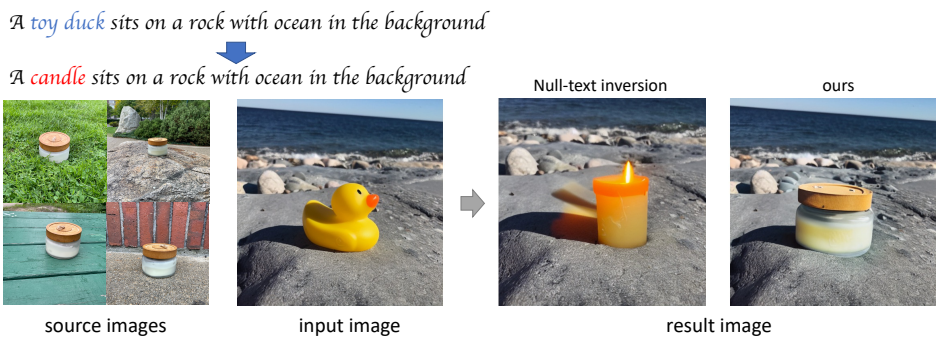


Figure 5.5: Results for personalized subject replacement.

1000 steps. The four proposed techniques make steady improvements over the baseline in DINO score while maintain CLIP-T score. The method utilizing all of our proposed techniques makes a remarkable 9.8-point improvement in DINO score over the baseline. For HiFi Tuner based on DreamBooth, we fine-tune all the diffusion model weights with 400 steps. By utilizing the reference-guided generation, HiFi Tuner achieves a 1.8-point improvement over the baseline in DINO score.

Results for personalized subject replacement. We show the qualitative results in Fig. 5.5. More results can be found in Appendix B.2. Personalized subject replacement is a new task we proposed in this paper. Null-text inversion is not able to substitute the subject in an image with user-provided subject, as shown in Fig. 5.5. Image inpainting methods are not capable of generating a subject with variations in accordance with the image background. Therefore, they are not directly comparable with our method.

5.4 Discussions

Time complexity. First, there exists a trade-off between the sample fidelity and the time cost. Current fine-tuning strategies struggle to preserve the details of subjects. Even DreamBooth cannot perfectly preserve the structural layout of a subject as shown in Fig. 5.4. Therefore, we think it is reasonable to propose a viable means to improve the sample fidelity first. Second, our method consumes reasonable time for the fine tuning process. We only fine tune the framework with 650 steps for the subject textual representations or 400 steps for diffusion model weights. This is much shorter than Textual Inversion (5000 steps) or DreamBooth (1000 steps). However, our method takes more time ($\sim$2 minutes on A100) for

Reference-Guided Generation which is on-par with the null-text inversion method.

Success rate of generation: It is worth mentioning that it lacks a well-recognized metric in measuring the success rate to the best of our knowledge, because the judge of the quality of the generated image is quite subjective. However, our method builds on top of either Textual Inversion or DreamBooth; according to our empirical experiments, we do not find obvious difference in success rate of generation comparing our method to Textual Inversion or DreamBooth separately.

5.5 Conclusions

In this work, we introduce a parameter-efficient fine-tuning method that can boost the sample fidelity and the prompt fidelity based on either Textual Inversion or DreamBooth. We propose step-wise subject representations comprising mask guidance and parameter regulations to improve the sample fidelity. We invents a reference-guided generation technique to mitigate the unwanted variations and artifacts for the generated subjects. We also exemplify that our method can be extended to substitute a subject in an image with personalized item by textual manipulations.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

In this dissertation, we have embarked on an in-depth exploration of Vision-Language Models (VLMs), focusing on three critical aspects: reasoning, scaling, and generating. Our journey through the intricate landscape of VLMs has revealed both the vast potential and the significant challenges inherent in integrating visual and linguistic information to create models that can understand, interpret, and generate human-like responses.

Reasoning with VLMs has shown us the importance of developing models capable of complex cognitive processes, such as inference and deduction, within multimodal contexts. We create a framework that bridges the gap between the high-dimensional, continuous data of visual inputs and the discrete, symbolic reasoning processes that characterize human cognitive capabilities. By introducing an induced symbolic space, our method enables the extraction and manipulation of symbolic representations directly from raw visual data, thereby facilitating a form of interpretable visual reasoning that is both robust and adaptable across diverse tasks. We demonstrated through extensive experiments that our approach not only achieves competitive performance on benchmark datasets but also provides insights into the reasoning process through the interpretability of the induced symbolic space. This interpretability allows for an intuitive understanding of the model's decision-making process, making it more transparent and trustworthy, especially in critical applications where explainability is paramount.

Scaling VLMs has been another cornerstone of our exploration. As we push the boundaries of what these models can achieve, the computational and data requirements have grown exponentially. Our work has contributed to more efficient and effective scaling methods, ensuring that the growth in model capabilities is sustainable and accessible. Specifically, we introduce a groundbreaking approach to tackling the challenges inherent in video localization tasks, including moment retrieval, temporal action localization and temporal segmentation. By presenting a unified framework that leverages shared structures and methodologies across

these tasks, we have demonstrated not only the feasibility of such an approach but also its effectiveness in improving performance and efficiency across a variety of benchmarks. Our extensive experiments and evaluations reveal that UnLoc significantly outperforms existing methods in accuracy, scalability, and adaptability, highlighting the benefits of a unified approach to video localization. The framework's ability to seamlessly integrate with different video analysis tasks and its flexibility in accommodating various types of input data make it a versatile tool for researchers and practitioners alike.

Generating with VLMs has perhaps been the most visually captivating aspect of our research. We have delved into the creative potential of VLMs with diffusion models, harnessing their ability to produce rich, coherent, and contextually relevant visual contents. In this dissertation, we introduce a novel fine-tuning framework designed to significantly enhance the performance of diffusion models for generating high-fidelity outputs tailored to specific subjects according to textual prompts. This approach incorporates mask guidance, a novel parameter regularization and a step-wise subject representation learning strategy to improve the subject fidelity. Plus, we propose a reference-guided generation method to mitigate the unwanted artifacts for the generated subjects. Through rigorous testing, HiFi Tuner has demonstrated superior performance compared to existing methods, showcasing its ability to produce highly accurate and subject-relevant results. The work emphasizes the framework's potential to transform the customization and application of generative models, paving the way for future advancements in tailored content creation and problem-solving in digital environments.

Looking forward, the potential applications of VLMs are vast and varied, ranging from enhancing accessibility with automated content generation to providing sophisticated tools for education, entertainment, and beyond. To solidify the use of VLMs, we suggest the following future works to be done.

1. **Improved Multimodal Fusion Techniques:** Developing more sophisticated methods for integrating visual and textual information can enhance the model's understanding and generation capabilities. Future research could explore deeper, more complex architectures that allow for a more seamless and effective fusion of modalities.

2. **Enhanced Interpretability and Explainability:** While VLMs can achieve impressive performance, understanding how these models make decisions is crucial, especially for applications in sensitive areas. Research focused on making VLMs more interpretable and explainable to humans can increase trust and facil-

itate wider adoption.

3. **Broader cross-modality Generation:** Exploring advanced techniques for cross-modality generation can open up new possibilities for creative and practical applications. For example, develop a unified model capable of generating videos, images, texts with user-defined prompts as the users wish.

4. **Efficient Training and Inference:** As VLMs grow in complexity, their computational requirements also increase. Research into more efficient training and inference methods can help reduce the environmental impact and make these models more accessible to researchers and practitioners with limited resources.

As we chart more powerful VLMs, we must also be vigilant about the ethical implications and strive to ensure that these technologies are developed and deployed in a manner that is beneficial and equitable for all.

In conclusion, this dissertation contributes to the field of AI by advancing our understanding of reasoning, scaling, and generating with Vision-Language Models. It lays a foundation for future research and application, highlighting the incredible potential of these models to transform our interaction with technology and, by extension, the world around us. The journey ahead is both exciting and daunting, but with continued innovation, collaboration, and consideration, we can navigate the challenges and harness the power of VLMs to create a better future.

APPENDIX A

APPENDIX TO INTERPRETABLE VISUAL REASONING VIA INDUCED SYMBOLIC SPACE

A.1 Details of compositional reasoning frameworks

Baseline visual reasoning framework The original compositional reasoning framework [5] is similar to the phase 1 of our framework in Figure 2 of the main paper, except that it works on pixel-level instead of object-level features. To generate vs, it feeds the image to a ResNet101 [76] pretrained on ImageNet [78] and flatten the last feature maps across the width and height as vs. For the question inputs, we first convert each question word to its word embedding vector (ws), then input ws to a bidirectional LSTM [164, 165] to extract the question embedding vector q. The compositional reasoning module takes vs, ws and q as inputs and performs multi-step reasoning to attain m, the final step memory output. Finally, the classifier outputs the probability for each answer choice with a linear classifier over the concatenation of m and q.

The MAC reasoning module At each step, the i-th MAC cell receives the control signal c_{i-1} and the memory output from the previous step, m_{i-1}, and outputs the new memory vector m_i. The control unit computes the single c_i to control reading of vs in the R/W unit. Specifically, it computes the interactions among c_{i-1}, q_i, and each vector in ws to produce the attention weights, and weighted averages ws to produce c_i. The control unit of each MAC cell has a unique question embedding projection layer, while all other layers are shared. The R/W unit aims to read the useful vs and store the read information into m_i. It first computes the interactions among m_{i-1}, c_{i-1} and each vector in vs to attain the attention weights, weighted averages vs to produce a read vector r_i, and finally computes the interaction of r_i and m_{i-1} to produce m_i. The weights of the R/W units are shared across all MAC cells. The initial control signal and memory c_0 and m_0 are learnable parameters.

A.2 Implementation details

CLEVR We set the hidden dimension D to 512 in all modules. We follow [5] to design the question embedding module, the compositional module and the classifier. For the object-level feature extracter, we make the backbone ResNet34 learnable and zero-pad the output vs to 12 vectors in total for any image. Notice that the maximum number of objects in an image is 11, so that the reasoning module is able to read nothing into the memory for some steps. For the concept projection module, to cover the full view of vs, the conv1D consists of five 1D convolution layers with kernel sizes (7,5,5,5,5), each followed by a Batch Norm layer [166] and an ELU activation layer [167].

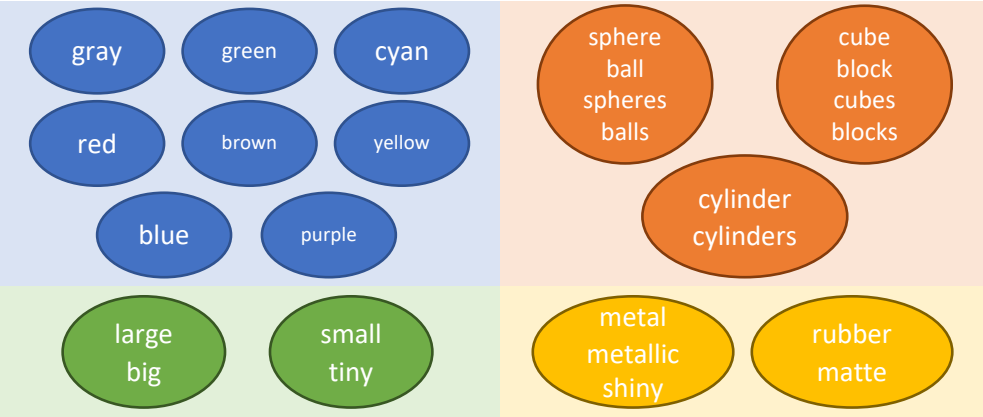
We use Adam optimizer [160] with momentum 0.9 and 0.999. Phase 1 and phase 2 share a same training schedule: the learning rate is initiated with 10^{-4} for the first 20 epochs and is halved every 5 epochs afterwards until stopped at the 40th epoch. We train the concept regression module separately with learning rate of 10^{-4} for 6 epochs. All the training process is conducted with a batch size of 256.

GQA The implementation details in the GQA setting basically follows the details on CLEVR. To better handle the complexity in GQA, we concatenate the object features with their corresponding bounding box coordinates to enhance the objects' location representations similar to [82]. We use GloVe [168] to initialize question word embeddings and maintain an exponential moving average with a decay rate of 0.999 to update the model parameters.

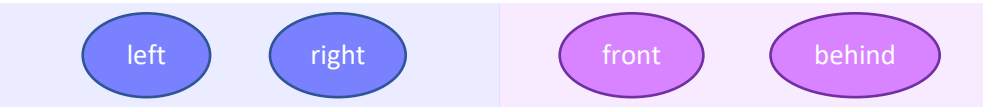
A.3 Visualization of the induced concept hierarchy

After visual mapping, binary coding and concept/super-concept induction, the unary concepts and super concepts are induced as shown in Figure A.1; the binary concepts are 'left', 'right', 'front' and 'behind', and {'left', 'right'} and {'front', 'behind'} form two super concept sets.

The generated concept hierarchy perfectly recovers the definition in CLEVR data generator and matches human prior knowledge, showing the success of our approach.



(a) unary concepts/super-concepts



(b) binary concepts/super-concepts

Figure A.1: Concepts and super concept sets. Each circle represents a concept described by the words in that circle. A super concept set comprises the concepts represented by circles of the same color.

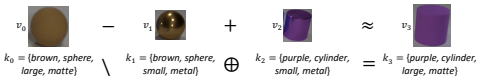


Figure A.2: An multi-modal analogy example enabled by our results.

A.4 Multi-modal concept analogy

Our concept induction results bridge the visual and symbolic spaces. The results enable to extend word analogy [90] (e.g., "Madrid" - "Spain" + "France" $\rightarrow$ "Paris") into the multi-modality setting. Figure A.2 gives an example, starting with the initial object v_0 and its predicted concepts K_0, subtracting concepts K_1 and adding new concepts K_2 result in a new concept set K_3 (Figure A.2 (bottom)). Then, if we retrieve visual object v_i with each concept set K_i along the path (Figure A.2 (top)), we have $v_0 - v_1 + v_2 \approx v_3$ in the original visual feature space.

A.5 Derivation from the concept interpretation

With the induced concepts and super concept sets, each object can be represented with a zero-one vector, k, where the entry is 1 if that object possesses the cor-

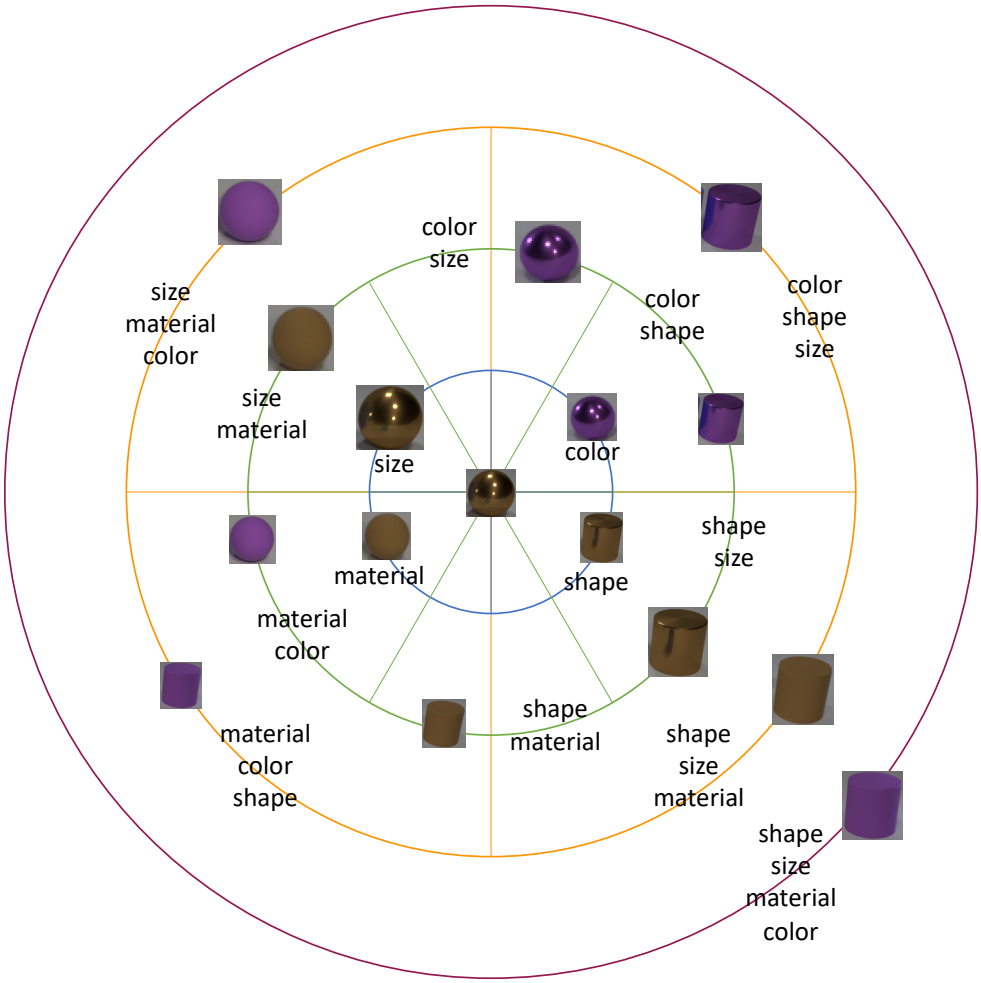


Figure A.3: Illustration of the semantic distance.

responding concept or 0 otherwise. Notice that the super concept sets split the whole concept set; we thereby name the entries of k corresponding to one super concept set as a super concept. The super concept is thus a zero-one vector with exactly one entry to be 1. We name this pattern as the super concept constraint. Therefore, we can define the semantic distance between two visual objects by the number of different super concepts or by Eqn. (A.1).

$$\zeta_{k_1,k_2} = \frac{|k_1 \oplus k_2|_1^1}{2}, \tag{A.1}$$

where k_1 and k_2 are the concept vectors representing two objects and $\oplus$ is the operation XOR. Studying the concepts and super concept sets induced, we acknowledge that the super concept sets correspond to color, shape, size and material in semantics. Thereby, we give an example of the semantic distances of multiple ob-

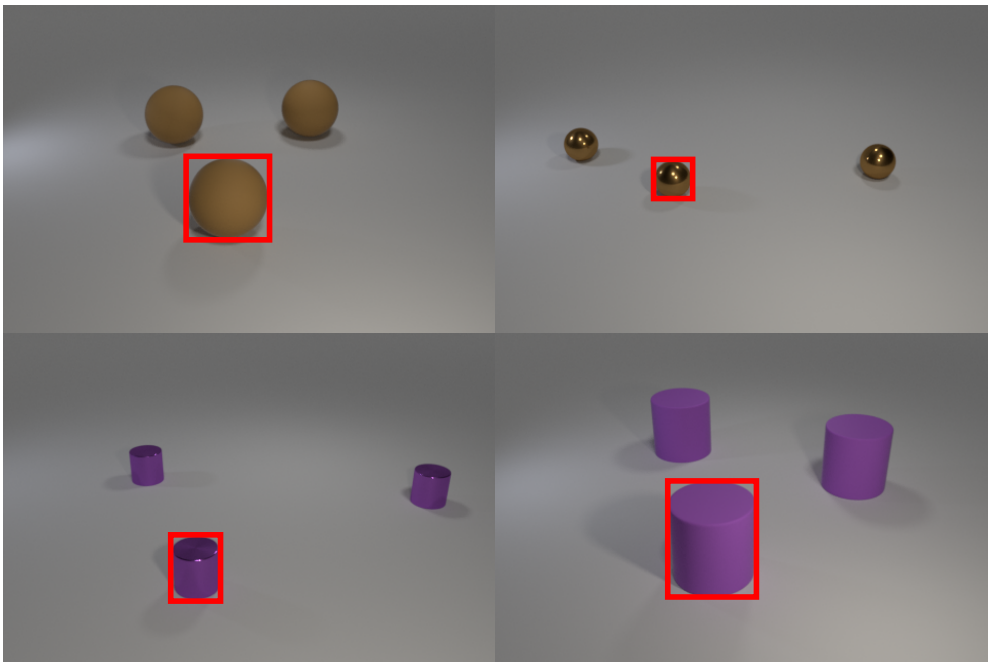


Figure A.4: The original images for extracting visual features. The object-level features corresponding to the objects bounded by red rectangles are used for the illustration of semantic operations in the visual feature space.

jects to one object, as shown in Figure A.3. The circle radii indicate the semantic distances to the object at the centers of these circles. The inner three circles are segmented so that each segment represents what super concepts are different. The outer circle represents all the 4 super concepts are different between the object on that circle and the object at the center.

We can further interpret the semantic analogy in the visual feature space with the induced concept vectors. Shown in Figure A.4, we first generate four images of different objects; then, we use our trained OCCAM structure to extract the object-level features corresponding to the objects bounded by red rectangles. Shown in Figure A.5(a), we can move the visual feature vector of the leftmost object closer to that of the rightmost object by subtracting and adding visual feature vectors of two other objects. The proximity between pairs of visual feature vectors is measured with cosine similarity as shown in Figure A.5(b). In the concept vector space, we can define a 'minus' operation, $k_1 \backslash k_2$, as eliminate the shared super concepts between k_1 and k_2 from k_1. We can also define a 'plus' operation, $k_1' \oplus k_2$, between a concept vector template k_1' and a concept vector k_2 as add the super concepts of o_2 that o_1' misses to o_1'. Therefore, the operations in the visual feature space can be explained with the operations we defined in the concept

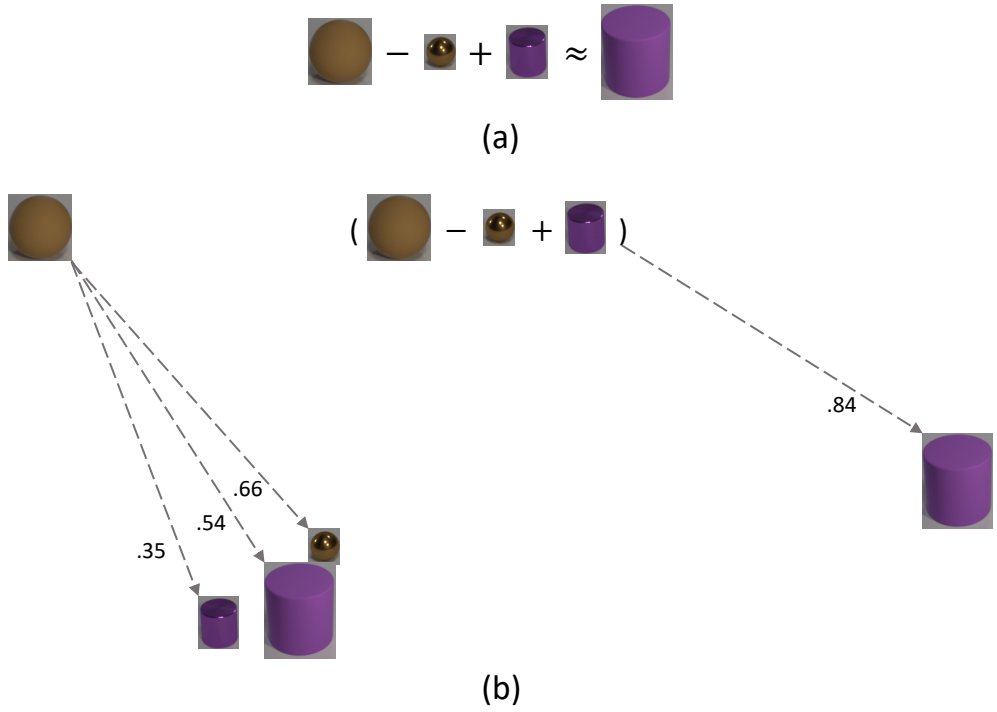


Figure A.5: Illustration of the semantic analogy in the visual feature space. (a) The operations on the visual features. (b) The cosine similarities between pairs of visual feature vectors.

vector space shown in Figure A.6.

A.6 Visualization of reasoning steps

We give an example of the compositional reasoning steps on the induced concept space of OCCAM, as shown in Figure A.7. While the attention is directly imposed on the projected concept vectors in the read unit of the compositional reasoning module, the attention can be equally mapped to the concept vectors and the visual objects as the projected concept vector to the concept vector or the projected concept vector to the visual object is a one-to-one mapping relationship. We also give an example of the compositional reasoning steps on the GQA dataset shown in Figure A.8. As the dimension of the induced concept vectors is too high, here we only present the attention on objects in the image.

gray	0		0		0		0
red	0		0		0		0
blue	0		0		0		0
green	0		0		0		0
brown	1		1		0		0
purple	0		0		1		1
cyan	0		0		0		0
yellow	0	\	0	⊕	0	=	0
sphere	1		1		0		0
cube	0		0		0		0
cylinder	0		0		1		1
large	1		0		0		1
small	0		1		1		0
metal	0		1		1		0
matte	1		0		0		1

Figure A.6: Operations on the concept vectors.

A.7 Human study

We assess the concept and super concept induction by studying how the word correlation conforms with our human knowledge. We present an extended subset of GQA concept correlations shown in Figure A.9. It consists of the 98 most common single words for describing objects. Each entry in the matrix represents the conditional probability that the column attribute exists given the row attribute exists. A pair of mutual high correlation values between two words indicates that these words belong to the same concept, while the opposite means that the concepts represented by those words belong to a super concept. Therefore, we can evaluate the concept induction by assessing the conditional probabilities of synonyms or uncorrelated words for each word, because from us human understanding, a synonym is used to describe the same concept while an uncorrelated word describes a concept belonging to the same super concept.

For each word in the extended subset words, we first let annotators choose 2 synonyms and 2 uncorrelated words from the rest 97 words. Then, rank the four chosen words in a descending order of similarity between them and the original word. Based on these annotations, we conduct two experiments: 1) measure the accuracy of classifying the chosen words to synonyms and uncorrelated words; 2) measure the Kendall tau distance [169] between the word similarity ranking based

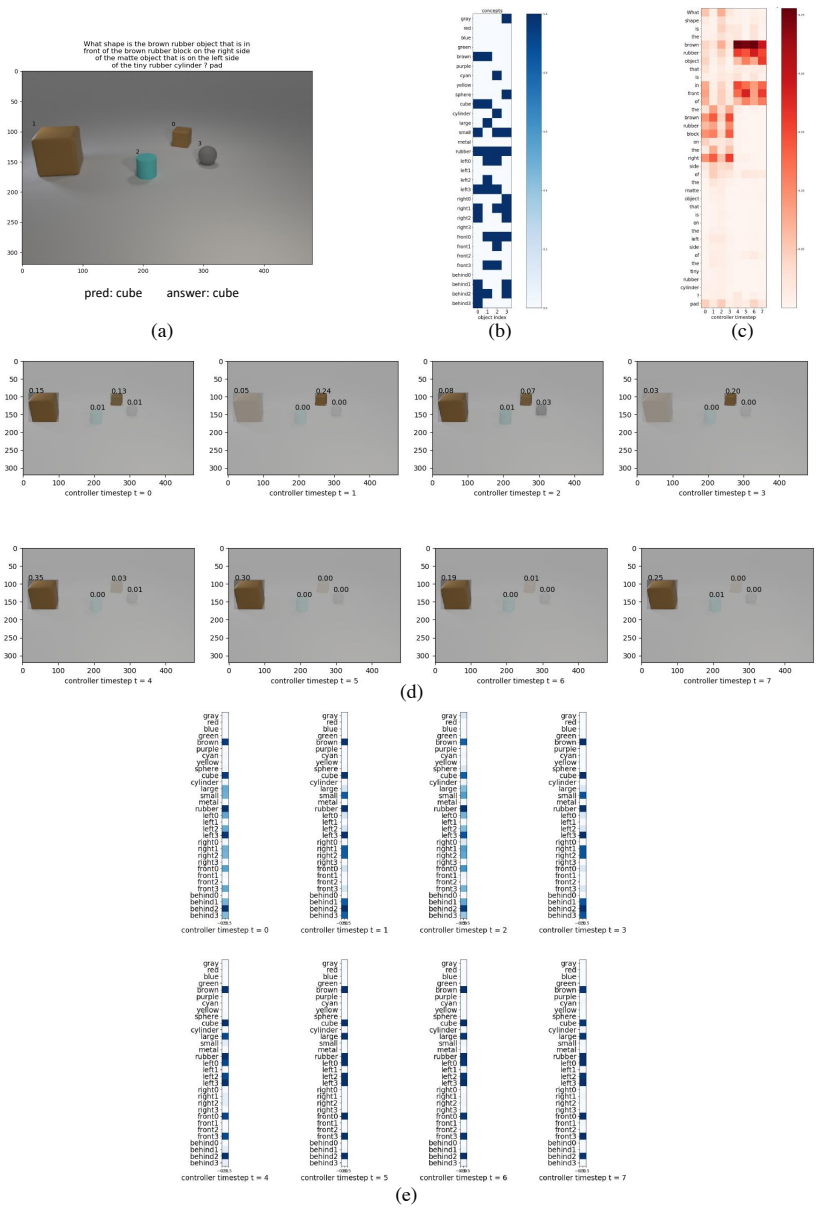


Figure A.7: Visualization of reasoning steps on CLEVR dataset. (a) The question, image, prediction and ground truth answer. The index of each object is shown on the upper left of the object. (b) The induced concepts of objects and relations. (c) The stepwise attentions on question words. (d) The stepwise attentions on objects. (e) The concept vector read into the memory of the reasoning module in each step.

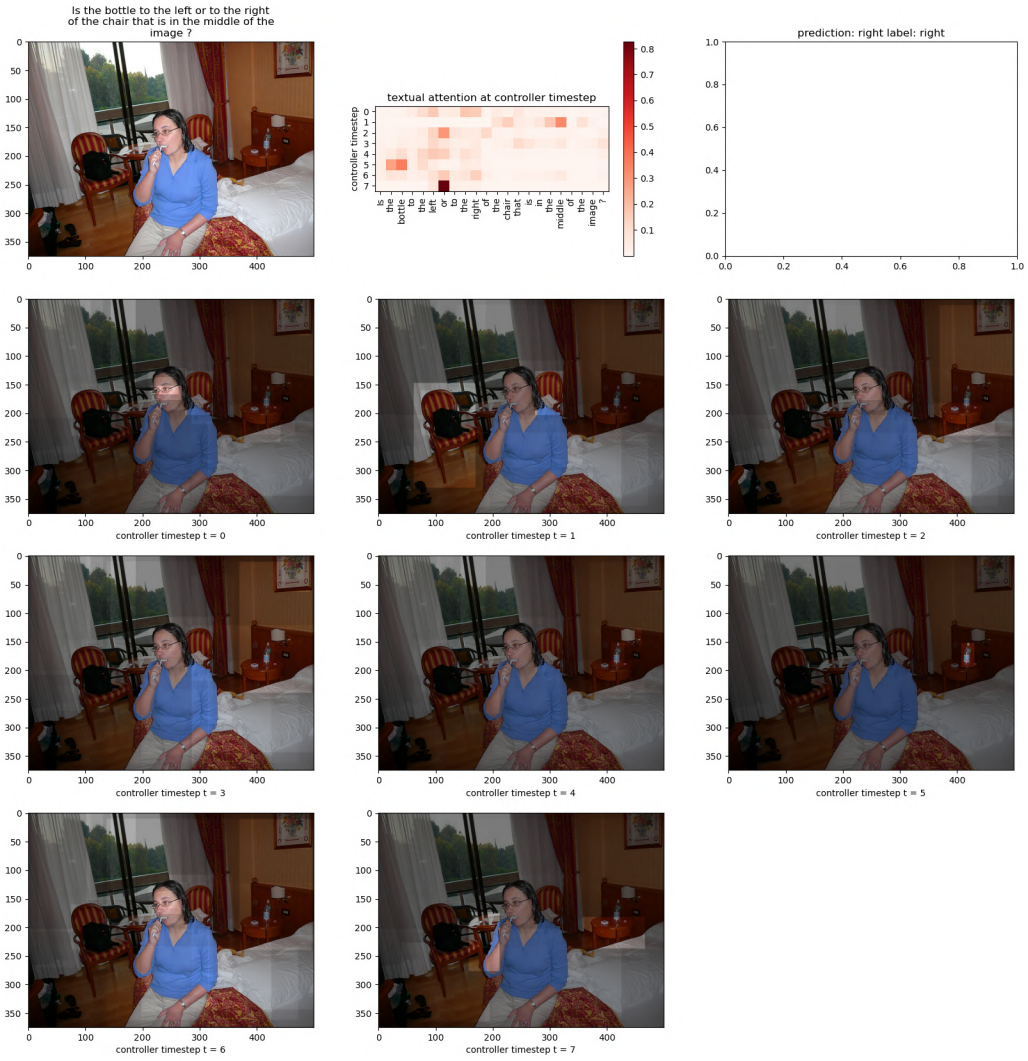


Figure A.8: Visualization of reasoning steps on GQA dataset.

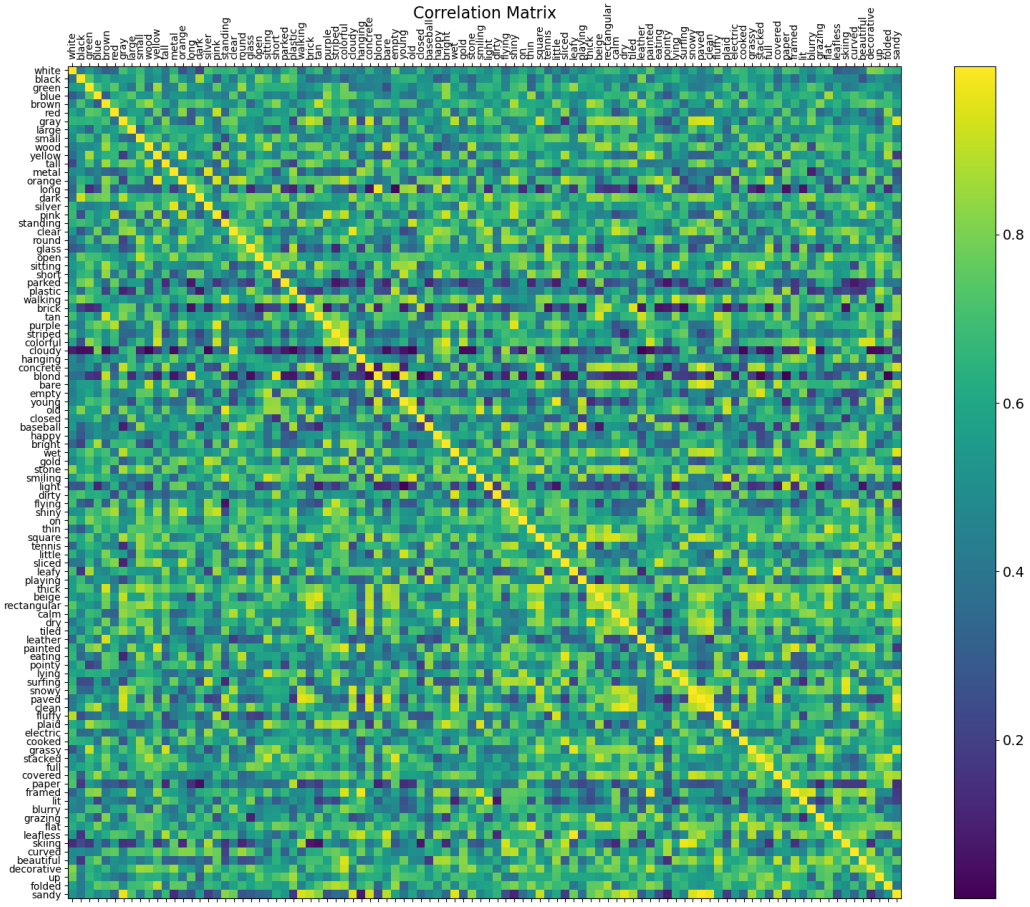


Figure A.9: The extended subset of GQA concept correlations.

Table A.1: The accuracy of classifying synonyms and uncorrelated words. A^{pos} represents the accuracy of classifying only synonyms. A^{neg} represents the accuracy of classifying only antonyms. †For word2vec, we tune the threshold on ground truth, while our method is used out of the box without threshold tuning (i.e., threshold set to 0.5).

Method	A^{pos}	A^{neg}	A
word2vec†	76.02%	60.71%	68.37%
induction	92.35%	63.78%	78.06%

on the conditional probability and that ranking based on human knowledge.

For the first experiment, we use a binary classifier with threshold 0.5 to classify the chosen words by humans. If a word's conditional probability given the original word is greater than the threshold, this word is classified as a synonym; if smaller, this word is classified as an uncorrelated word. The accuracy can be calculated with Eqn. (A.2).

$$A = \frac{1}{|S|}\sum_{i\in S}\frac{1}{|W_i|}(\sum_{j\in W_i^{pos}}\mathbb{1}(R_{i,j} > t) + \sum_{j\in W_i^{neg}}\mathbb{1}(R_{i,j} < t)), \tag{A.2}$$

where A represents accuracy, S is the subset of words, W_i represents the set of synonyms and uncorrelated words chosen for word i, $R_{i,j}$ represents the conditional probability of word j given word i exists and t is the threshold. For comparison, we also calculate the cosine similarity of word GloVe [168] embeddings to substitute the conditional probability and serve as R in Eqn. (A.2). For this setting, we tune the threshold t to be 0.21 to reach the best accuracy. The result in Table A.1 shows that our induction highly conforms with our human sense in grouping words into concepts but does not agree much with humans in grouping super concepts. By further studying specific cases, we realize that a word and its uncorrelated words defined by humans can simultaneously describe one object. For example, 'white' and 'black' can be used together to describe a zebra; 'leafy' and 'leafless' both describes a status of a plant. Such words have high correlations, which aligns with our human understanding.

The second experiment measures how the induced word proximity conforms with our human knowledge. For a word w_i, our annotators rank the chosen synonyms and uncorrelated words $a_i = (a_{i1}, a_{i2}, a_{i3}, a_{i4})$ with a descending order of word similarity to w_i and assign a sequence of order indices $O_i^{human} = (0, 1, 2, 3)$ to a_i. Then, we rank $(a_{i1}, a_{i2}, a_{i3}, a_{i4})$ with a descending order of their conditional probabilities and assign a sequence of order indices O_i^{induce} to a_i. For comparison,

Table A.2: The average ranking distance to human rankings.

$D(O^{word2vec})$	$D(O^{induce})$
0.3418	0.2585

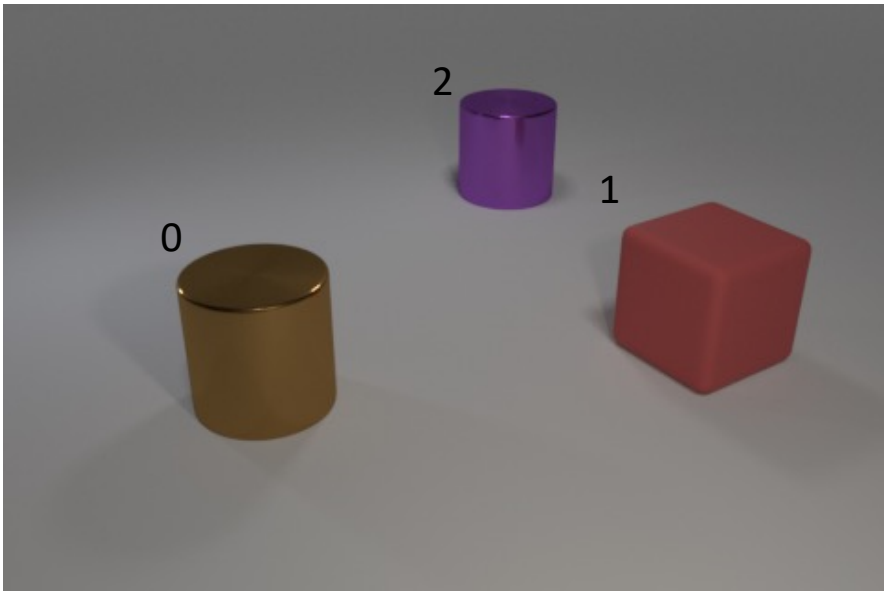
we further rank $(a_{i1}, a_{i2}, a_{i3}, a_{i4})$ in a descending order of cosine similarities between the GLoVe embeddings of $(a_{i1}, a_{i2}, a_{i3}, a_{i4})$ and w_i and assign a sequence of order indices $O_i^{word2vec}$ to a_i. The average ranking distance can be calculated with Eqn. (A.3).

$$D(O^x) = \frac{1}{|S|} \sum_{i \in S} \mathcal{K}(O_i^{human}, O_i^x), \tag{A.3}$$

where D represents the average ranking distance, $x \in \{induce, word2vec\}$, $\mathcal{K}$ represents the operation for calculating the normalized Kendall tau distance between two rankings. The result in Table (A.2) proves that our induction from visual language relations encodes word proximity that is more aligned with human knowledge than the one encoded by GloVe embeddings from language-only data.

A.8 Error analysis

The reasoning process may reach a false answer if 1) a concept is mentioned in the question and 2) that concept is wrongly classified for the objects ought to be attended. However, the reasoning process may still reach a correct answer if either of these two conditions is not met. We present two examples in Figure A.10.



	obj0	obj1	obj2
gray	0	0	0
red	0	1	0
blue	0	0	0
green	0	0	0
brown	1	0	0
purple	0	0	1
cyan	0	0	0
yellow	0	0	0
sphere	0	0	0
cube	0	1	0
cylinder	1	0	1
large	1	1	1
small	0	0	0
metal	1	0	1
matte	0	1	0

	obj0	obj1	obj2
left0	0	0	0
left1	1	0	1
left2	1	0	0
right0	0	1	1
right1	0	0	0
right2	0	1	0
front0	0	1	0
front1	1	0	0
front2	1	1	0
behind0	0	0	1
behind1	0	0	1
behind2	0	0	0

question	answer	ground truth
There is a big brown metal cylinder; how many large matte cubes are behind it?	0	1
What is the color of the rubber cube?	red	red

Figure A.10: Error analysis. The predicted unary and binary concepts corresponding to each object in the image above are shown in the tables at the middle; the digits colored in red are wrong predicted concepts. The questions, the predicted answers and the ground truth answers are shown in the table at the bottom.

APPENDIX B

APPENDIX TO HIFI TUNER: HIGH-FIDELITY SUBJECT-DRIVEN FINE-TUNING FOR DIFFUSION MODELS

B.1 Algorithm of optimizing T_θ

Please refer to Algorithm 6 for optimizing T_θ.

Algorithm 6: Algorithm of optimizing T_θ. $P(M) \in \mathbb{R}^{N\times 3}$ returns the coordinates where $M == 1$ and appends 1's after the coordinates.

Result: T_θ^*
Inputs: M_r, M_g
$P_r = P(M_r)$, $P_g = P(M_g)$
for $l = [1, ..., L]$ **do**
  $s = 0$
  $P_t = T_\theta(P_r)$
  for $p_t \in P_t$ **do**
    $m = MAX\_FLOAT$
    for $p_g \in P_g$ **do**
      $x = \|p_t - p_g\|_2^2$
      if $x < m$ **then**
        $m = x$
    $s = s + m$
  $\theta = \theta - \eta \nabla_\theta s$
$T_\theta^* = T_\theta$

B.2 Results for personalized subject replacement

We show more results for the personalized subject replacement in Figure B.1.

Figure B.1: Results for personalized subject replacement.

REFERENCES

[1] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 21–29.

[2] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 451–466.

[3] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6077–6086.

[4] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, "Film: Visual reasoning with a general conditioning layer," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[5] D. A. Hudson and C. D. Manning, "Compositional attention networks for machine reasoning," in *International Conference on Learning Representations (ICLR)*, 2018.

[6] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Learning to compose neural networks for question answering," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NACACL)*, 2016.

[7] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 39–48.

[8] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, "Learning to reason: End-to-end module networks for visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 804–813.

[9] J. Johnson, B. Hariharan, L. Van Der Maaten, J. Hoffman, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Inferring and executing programs for visual reasoning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2989–2998.

[10] R. Hu, J. Andreas, T. Darrell, and K. Saenko, "Explainable neural computation via stack neural module networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 53–69.

[11] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 1031–1042.

[12] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *International Conference on Learning Representations (ICLR)*, 2018.

[13] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 670–685.

[14] M. Bajaj, L. Wang, and L. Sigal, "G3raphground: Graph-based language grounding," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 4281–4290.

[15] R. A. Yeh, M. N. Do, and A. G. Schwing, "Unsupervised textual grounding: Linking words to image concepts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6125–6134.

[16] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6541–6549.

[17] Q. Zhang, Y. Yang, H. Ma, and Y. N. Wu, "Interpreting cnns via decision trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6261–6270.

[18] J. Shi, H. Zhang, and J. Li, "Explainable and explicit visual reasoning over scene graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8376–8384.

[19] D. H. Park, L. A. Hendricks, Z. Akata, B. Schiele, T. Darrell, and M. Rohrbach, "Attentive explanations: Justifying decisions and pointing to the evidence," *arXiv preprint arXiv:1612.04757*, 2016.

[20] T. Lei, R. Barzilay, and T. Jaakkola, "Rationalizing neural predictions," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 107–117.

[21] J. Chen, L. Song, M. Wainwright, and M. Jordan, "Learning to explain: An information-theoretic perspective on model interpretation," in *International Conference on Machine Learning (ICML)*, 2018, pp. 883–892.

[22] M. Yu, S. Chang, Y. Zhang, and T. Jaakkola, "Rethinking cooperative rationalization: Introspective extraction and complement control," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 4085–4094.

[23] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," 2014.

[24] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3128–3137.

[25] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1–9.

[26] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2425–2433.

[27] C. Han, J. Mao, C. Gan, J. Tenenbaum, and J. Wu, "Visual concept-metaconcept learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 5001–5012.

[28] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1049–1058.

[29] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1130–1139.

[30] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2914–2923.

[31] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "Bmn: Boundary-matching network for temporal action proposal generation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3889–3898.

[32] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[33] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles, "End-to-end, single-stream temporal action detection in untrimmed videos," *British Machine Vision Association and Society for Pattern Recognition*, 2019.

[34] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 988–996.

[35] M. Nawhal and G. Mori, "Activity graph transformer for temporal action localization," *arXiv preprint arXiv:2101.08540*, 2021.

[36] C.-L. Zhang, J. Wu, and Y. Li, "Actionformer: Localizing moments of actions with transformers," in *European Conference on Computer Vision*. Springer, 2022, pp. 492–510.

[37] C. Ju, T. Han, K. Zheng, Y. Zhang, and W. Xie, "Prompting visual-language models for efficient video understanding," in *European Conference on Computer Vision*. Springer, 2022, pp. 105–124.

[38] S. Nag, X. Zhu, Y.-Z. Song, and T. Xiang, "Zero-shot temporal action detection via vision-language prompting," in *European Conference on Computer Vision*. Springer, 2022, pp. 681–697.

[39] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, "Localizing moments in video with temporal language," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[40] J. Gao, C. Sun, Z. Yang, and R. Nevatia, "Tall: Temporal activity localization via language query," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5267–5275.

[41] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua, "Temporally grounding natural sentence in video," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 162–171.

[42] J. Wang, L. Ma, and W. Jiang, "Temporally grounding language queries in videos by contextual boundary-aware prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 168–12 175.

[43] Y. Yuan, L. Ma, J. Wang, W. Liu, and W. Zhu, "Semantic conditioned dynamic modulation for temporal sentence grounding in videos," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[44] L. Chen, C. Lu, S. Tang, J. Xiao, D. Zhang, C. Tan, and X. Li, "Rethinking the bottom-up framework for query-based video localization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 10 551–10 558.

[45] R. Zeng, H. Xu, W. Huang, P. Chen, M. Tan, and C. Gan, "Dense regression network for video grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 287–10 296.

[46] S. Zhang, H. Peng, J. Fu, and J. Luo, "Learning 2d temporal adjacent networks for moment localization with natural language," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 870–12 877.

[47] J. Mun, M. Cho, and B. Han, "Local-global video-text interactions for temporal grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 810–10 819.

[48] J. Lei, T. L. Berg, and M. Bansal, "Detecting moments and highlights in videos via natural language queries," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 846–11 858, 2021.

[49] Y. Liu, S. Li, Y. Wu, C.-W. Chen, Y. Shan, and X. Qie, "Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3042–3051.

[50] B. Zhang, H. Hu, J. Lee, M. Zhao, S. Chammas, V. Jain, E. Ie, and F. Sha, "A hierarchical multi-modal encoder for moment localization in video corpus," *arXiv preprint arXiv:2011.09046*, 2020.

[51] J. Gao and C. Xu, "Fast video moment retrieval," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1523–1532.

[52] Y. Tang, D. Ding, Y. Rao, Y. Zheng, D. Zhang, L. Zhao, J. Lu, and J. Zhou, "Coin: A large-scale dataset for comprehensive instructional video analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1207–1216.

[53] C. Sun, F. Baradel, K. Murphy, and C. Schmid, "Learning video representations using contrastive bidirectional transformer," *arXiv preprint arXiv:1906.05743*, 2019.

[54] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9879–9889.

[55] H. Xu, G. Ghosh, P.-Y. Huang, D. Okhonko, A. Aghajanyan, F. Metze, L. Zettlemoyer, and C. Feichtenhofer, "Videoclip: Contrastive pre-training for zero-shot video-text understanding," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6787–6800.

[56] H. Luo, L. Ji, B. Shi, H. Huang, N. Duan, T. Li, J. Li, T. Bharti, and M. Zhou, "Univl: A unified video and language pre-training model for multimodal understanding and generation," *arXiv preprint arXiv:2002.06353*, 2020.

[57] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2630–2640.

[58] A. Casanova, M. Careil, J. Verbeek, M. Drozdzal, and A. Romero Soriano, "Instance-conditioned gan," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 517–27 529, 2021.

[59] Y. Nitzan, K. Aberman, Q. He, O. Liba, M. Yarom, Y. Gandelsman, I. Mosseri, Y. Pritch, and D. Cohen-Or, "Mystyle: A personalized generative prior," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–10, 2022.

[60] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[61] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2022, pp. 10 684–10 695.

[62] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans et al., "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 36 479–36 494, 2022.

[63] W. Chen, H. Hu, C. Saharia, and W. W. Cohen, "Re-imagen: Retrieval-augmented text-to-image generator," in *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. [Online]. Available: https://openreview.net/forum?id=XSEBx0iSjFQ

[64] S. Sheynin, O. Ashual, A. Polyak, U. Singer, O. Gafni, E. Nachmani, and Y. Taigman, "Knn-diffusion: Image generation via large-scale retrieval," *arXiv preprint arXiv:2204.02849*, 2022.

[65] W. Chen, H. Hu, Y. Li, N. Rui, X. Jia, M.-W. Chang, and W. W. Cohen, "Subject-driven text-to-image generation via apprenticeship learning," *arXiv preprint arXiv:2304.00186*, 2023.

[66] X. Jia, Y. Zhao, K. C. Chan, Y. Li, H. Zhang, B. Gong, T. Hou, H. Wang, and Y.-C. Su, "Taming encoder for zero fine-tuning image customization with text-to-image diffusion models," *arXiv preprint arXiv:2304.02642*, 2023.

[67] J. Shi, W. Xiong, Z. Lin, and H. J. Jung, "Instantbooth: Personalized text-to-image generation without test-time finetuning," *arXiv preprint arXiv:2304.03411*, 2023.

[68] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 22 500–22 510.

[69] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-or, "An image is worth one word: Personalizing text-to-image generation using textual inversion," in *International Conference on Learning Representations (ICLR)*, 2023. [Online]. Available: https://openreview.net/forum?id=NAQvF08TcyG

[70] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "Styleclip: Text-driven manipulation of stylegan imagery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2085–2094.

[71] O. Avrahami, D. Lischinski, and O. Fried, "Blended diffusion for text-driven editing of natural images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 18 208–18 218.

[72] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "SDEdit: Guided image synthesis and editing with stochastic differential equations," in *International Conference on Learning Representations (ICLR)*, 2022.

[73] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-or, "Prompt-to-prompt image editing with cross-attention control," in *International Conference on Learning Representations (ICLR)*, 2023.

[74] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or, "Null-text inversion for editing real images using guided diffusion models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 6038–6047.

[75] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 392–18 402.

[76] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[77] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[78] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[79] G. Xuan, W. Zhang, and P. Chai, "Em algorithms of gaussian mixture model and hidden markov model," in *International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2001, pp. 145–148.

[80] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4967–4976.

[81] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 5103–5114.

[82] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko, "Language-conditioned graph networks for relational reasoning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 10 294–10 303.

[83] W. Chen, Z. Gan, L. Li, Y. Cheng, W. Wang, and J. Liu, "Meta module network for compositional visual reasoning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 655–664.

[84] D. A. Hudson and C. D. Manning, "Learning by abstraction: The neural state machine," *arXiv preprint arXiv:1907.03950*, 2019.

[85] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," *arXiv preprint arXiv:2102.03334*, 2021.

[86] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2901–2910.

[87] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma et al., "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision (IJCV)*, vol. 123, no. 1, pp. 32–73, 2017.

[88] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6700–6709.

[89] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[90] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013, pp. 3111–3119.

[91] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning (ICML)*. PMLR, 2021, pp. 8748–8763.

[92] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*. PMLR, 2021, pp. 4904–4916.

[93] B. Ni, H. Peng, M. Chen, S. Zhang, G. Meng, J. Fu, S. Xiang, and H. Ling, "Expanding language-image pretrained models for general video recognition," in *European Conference on Computer Vision*. Springer, 2022, pp. 1–18.

[94] Z. Lin, S. Geng, R. Zhang, P. Gao, G. De Melo, X. Wang, J. Dai, Y. Qiao, and H. Li, "Frozen clip models are efficient video learners," in *European Conference on Computer Vision*. Springer, 2022, pp. 388–404.

[95] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen et al., "Simple open-vocabulary object detection with vision transformers. arxiv 2022," *arXiv preprint arXiv:2205.06230*, vol. 2, 2022.

[96] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, "Scaling open-vocabulary image segmentation with image-level labels," in *European Conference on Computer Vision*. Springer, 2022, pp. 540–557.

[97] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "A clip-hitchhiker's guide to long video retrieval," *arXiv preprint arXiv:2205.08508*, 2022.

[98] M. Soldan, M. Xu, S. Qu, J. Tegner, and B. Ghanem, "Vlg-net: Video-language graph matching network for video grounding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3224–3234.

[99] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, "Dense-captioning events in videos," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 706–715.

[100] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2015, pp. 961–970.

[101] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, "The thumos challenge on action recognition for videos "in the wild"," *Computer Vision and Image Understanding*, vol. 155, pp. 1–23, 2017.

[102] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 244–253.

[103] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European Conference on Computer Vision*. Springer, 2022, pp. 280–296.

[104] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[105] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[106] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5561–5569.

[107] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[108] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in neural information processing systems*, vol. 34, pp. 12 116–12 128, 2021.

[109] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[110] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological cybernetics*, vol. 20, no. 3, pp. 121–136, 1975.

[111] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[112] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 993–13 000.

[113] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, "G-tad: Sub-graph localization for temporal action detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 156–10 165.

[114] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the kinetics-700 human action dataset," *arXiv preprint arXiv:1907.06987*, 2019.

[115] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6836–6846.

[116] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[117] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European Conference on Computer Vision*. Springer, 2016, pp. 646–661.

[118] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357.

[119] X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer, "Lit: Zero-shot transfer with locked-image text tuning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 123–18 133.

[120] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, "Span-based localizing network for natural language video localization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6543–6554.

[121] G. Nan, R. Qiao, Y. Xiao, J. Liu, S. Leng, H. Zhang, and W. Lu, "Interventional video grounding with dual contrastive learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2765–2775.

[122] W. Moon, S. Hyun, S. Park, D. Park, and J.-P. Heo, "Query-dependent video representation for moment retrieval and highlight detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 023–23 033.

[123] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[124] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.

[125] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[126] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[127] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[128] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6202–6211.

[129] L. Yang, H. Peng, D. Zhang, J. Fu, and J. Han, "Revisiting anchor mechanisms for temporal action localization," *IEEE Transactions on Image Processing*, vol. 29, pp. 8535–8548, 2020.

[130] H. Alwassel, S. Giancola, and B. Ghanem, "Tsp: Temporally-sensitive pretraining of video encoders for localization tasks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3173–3183.

[131] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 344–353.

[132] C. Zhao, A. K. Thabet, and B. Ghanem, "Video self-stitching graph network for temporal action localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 658–13 667.

[133] X. Liu, Q. Wang, Y. Hu, X. Tang, S. Zhang, S. Bai, and X. Bai, "End-to-end temporal action detection with transformer," *IEEE Transactions on Image Processing*, vol. 31, pp. 5427–5441, 2022.

[134] Q. Liu and Z. Wang, "Progressive boundary refinement network for temporal action detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 612–11 619.

[135] Z. Qing, H. Su, W. Gan, D. Wang, W. Wu, X. Wang, Y. Qiao, J. Yan, C. Gao, and N. Sang, "Temporal context aggregation network for temporal action proposal refinement," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 485–494.

[136] Z. Zhu, W. Tang, L. Wang, N. Zheng, and G. Hua, "Enriching local and global contexts for temporal action localization," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 13 516–13 525.

[137] L. Zhu and Y. Yang, "Actbert: Learning global-local video-text representations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8746–8755.

[138] J. Yang, Y. Bisk, and J. Gao, "Taco: Token-aware cascade contrastive learning for video-text alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 562–11 572.

[139] H. Xu, G. Ghosh, P.-Y. Huang, P. Arora, M. Aminzadeh, C. Feichtenhofer, F. Metze, and L. Zettlemoyer, "Vlm: Task-agnostic video-language model pre-training for video understanding," *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021.

[140] J. Huh, J. Chalk, E. Kazakos, D. Damen, and A. Zisserman, "Epic-sounds: A large-scale dataset of actions that sound," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[141] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[142] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems (NeurIPS)*, vol. 33, pp. 6840–6851, 2020.

[143] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[144] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman et al., "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 25 278–25 294, 2022.

[145] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer et al., "Pali: A jointly-scaled multilingual language-image model," in *The Eleventh International Conference on Learning Representations (ICLR)*, 2022.

[146] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[147] N. Ruiz, Y. Li, V. Jampani, W. Wei, T. Hou, Y. Pritch, N. Wadhwa, M. Rubinstein, and K. Aberman, "Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models," 2023.

[148] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[149] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[150] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations (ICLR)*, 2020.

[151] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2021, pp. 10 012–10 022.

[152] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu et al., "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[153] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, 2021, pp. 9650–9660.

[154] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[155] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning (ICML)*, 2023.

[156] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 461–11 471.

[157] S. Xie, Z. Zhang, Z. Lin, T. Hinz, and K. Zhang, "Smartbrush: Text and shape guided object inpainting with diffusion model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 22 428–22 437.

[158] X. Zhang, J. Guo, P. Yoo, Y. Matsuo, and Y. Iwasawa, "Paste, inpaint and harmonize via denoising: Subject-driven image editing with pre-trained diffusion model," *arXiv preprint arXiv:2306.07596*, 2023.

[159] T. Li, M. Ku, C. Wei, and W. Chen, "Dreamedit: Subject-driven image editing," *arXiv preprint arXiv:2306.12624*, 2023.

[160] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[161] S. Hao, K. Han, S. Zhao, and K.-Y. K. Wong, "Vico: Plug-and-play visual condition for personalized text-to-image generation," 2023.

[162] D. Li, J. Li, and S. C. H. Hoi, "Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing," 2023.

[163] X. Chen, L. Huang, Y. Liu, Y. Shen, D. Zhao, and H. Zhao, "Anydoor: Zero-shot object-level image customization," *arXiv preprint arXiv:2307.09481*, 2023.

[164] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[165] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[166] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, 2015.

[167] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *International Conference on Learning Representations (ICLR)*, 2016.

[168] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[169] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.