Connecting Language to Video Representation Learning

Thesis proposal presented

by

Kanchana Ranasinghe

to

The Graduate School

in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

June 2025

Abstract

Humans understand their surrounding world processing multi-sensory signals, with visual perception, especially in the form of video, playing a dominant role. Leveraging naturally occurring patterns such as *spatio-temporal relationships*, *paired multi-sensory signals*, and *3D geometry*, humans build strong internal world models without specific supervision, that is under what we call *self-supervised* settings. Recent advances in language processing and computer vision have led to large language models (LLMs) and their multimodal variants, which are trained under similar self-supervised settings, to build certain degrees of their own internal world models. However, in the context of video understanding, these models are still far from human-level performance.

In this work, we first explore how traditional video self-supervised learning (SSL), where learning involves only the visual modality, can be connected to natural language for improved video understanding. We leverage existing strong language-image models (CLIP) to guide a video SSL process, learning not only strong representations but also unlocking zero-shot inference capabilities given the language alignment of features. Despite language alignment, we retain the traditional SSL paradigm, allowing learning with only videos having no dependency on video level labels or captions.

Our initial investigation focusses on video classification tasks, where ability of models to perform fine-grained spatio-temporal reasoning may often not be crucial. As a next step, we explore a more complex language-tied task: video question answering (QnA). Multi-modal LLMs (MLLMs) excel at these tasks, but often face challenges in correctly modeling spatio-temporal object relationships in videos (i.e. spatial reasoning). We investigate how explicit object localization supervision can improve spatial reasoning of MLLMs. Utilizing natural language based location representations, we seamlessly integrate localization supervision into MLLM training, learning stronger video representations with better spatial awareness. We next extend to explicit motion representations, following a similar natural language formulation, leading to further improvements in video understanding. Language alone however struggles to capture fine-grained motion cues, motivating exploration into structered intermediate motion representations. As a final step, we learn explicit language to structured motion representation mappings from internet-scale video-caption data. We use these mappings beyond video QnA into real world interaction in the form of robot control, taking a significant step towards human-level world-models capable of video understanding.

Contents

List of Figures

List of Tables

Publications

The work presented in this thesis proposal also appeared in the following peer-reviewed articles or preprints:

1. Self-supervised Video Transformer
   K Ranasinghe, M Naseer, S Khan, FS Khan, MS Ryoo
   CVPR 2022

2. Language-based Action Concept Spaces Improve Video Self-Supervised Learning
   K Ranasinghe, MS Ryoo
   NeurIPS 2023

3. Learning to Localize Objects Improves Spatial Reasoning in Visual-LLMs
   K Ranasinghe, SN Shukla, O Poursaeed, MS Ryoo, TY Lin
   CVPR 2024

4. Understanding Long Videos with Multimodal Language Models
   K Ranasinghe, X Li, K Kahatapitiya, MS Ryoo
   ICLR 2025

5. Pixel Motion as Universal Representation for Robot Control
   K Ranasinghe, X Li, C Mata, J Park, MS Ryoo
   arXiv 2025

Chapter 1

Introduction

1.1 Overview

Humans understand their world through a seamless integration of multi-sensory inputs, with visual perception—particularly in the form of video—playing a dominant role. Our ability to extract spatio-temporal relationships, synchronize audio-visual cues, and perceive 3D geometry enables us to build rich internal world models, even in the absence of explicit supervision. This self-supervised learning capability is fundamental to our capacity for abstraction, prediction, and reasoning across time and space.

Replicating this ability in machines has been a longstanding goal of computer vision and artificial intelligence. Recent advances in large language models (LLMs) and their multimodal variants have enabled impressive capabilities in static image understanding and language processing. However, progress in video understanding has lagged behind. Despite being trained on internet-scale data, current models often struggle with fine-grained temporal reasoning, spatial awareness of objects, and motion understanding—especially in tasks requiring causal inference or real world interaction. These limitations reflect the inherent complexity of video data, where semantic information is distributed across time and dependent on both object dynamics and contextual cues.

To address this gap, we propose leveraging natural language as a guiding prior for video representation learning, while adhering to certain degrees of self-supervised learning principles. Language is a powerful modality: it is compositional, universal (different models trained to output English language can communicate, as opposed to those generating embeddings), and closely aligned with human cognition. Our central thesis is that aligning video representations

with natural language can unlock more general, interpretable, and transferable models of video understanding. Importantly, this alignment should not require excessive human supervision for the learning process. Instead we aim to retain the self-supervised nature of human learning, by using natural language to guide the representation learning process.

Our work proceeds in four stages. We begin by aligning video representations with language using self-supervised learning (LSS), then enhance spatial reasoning in visual language models (LocVLM). We extend to long video understanding through language-based object motion modeling (MVU), and finally introduce structured motion representations (LangToMo) to bridge language, vision, and action for real world interaction tasks.

Stage 1: Language-Guided Video Self-Supervised Learning (LSS). We propose LSS, a novel self-supervised framework that adapts powerful image-language models like CLIP to the video domain without requiring any per-video labels or captions. LSS introduces concept spaces derived from action-related textual prompts and distills video representations into these spaces via multi-view consistency objectives. The result is a video model that retains CLIP's zero-shot capabilities while learning temporal patterns through video-only training.

Stage 2: Enhancing Spatial Reasoning in Multimodal Language Models (LocVLM). While multimodal LLMs show strong performance on tasks like video question answering, they often lack precise spatial reasoning. We address this by explicitly integrating spatial supervision—represented purely in language form as textual coordinates—into visual LLM training. Our approach avoids architectural modifications and relies solely on location-aware prompts, resulting in improved spatial understanding, reduced hallucination, and the novel ability to describe and localize arbitrary image regions within videos.

Stage 3: Extending to Long Video Understanding (MVU). We extend our framework to long video understanding by introducing a modular language-based formulation of object motion trajectories. Our Multimodal Video Understanding (MVU) framework extracts object-centric information—such as motion, spatial location, and context—across multiple frames and expresses this as natural language. This enables explicit motion modeling in any off-the-shelf language model without requiring re-training. MVU significantly improves performance on temporally extended and motion-focused tasks while offering interpretability and scalability across diverse video domains.

Stage 4: Structured Motion Representation via Language and Vision (LangToMo). Motion understanding presents a unique challenge: while inherently

tied to language (e.g., "reach," "move left"), motion representations also encode state-dependent information about the actor, task objectives, and environmental interactions—context that cannot be fully captured through language alone. We propose learning structured intermediate motion representations, conditioned jointly on language and video frames, to bridge this gap. Using large-scale video-caption data, we train models to map natural language commands to pixel-level motion, enabling applications such as zero-shot robotic control and interpretable motion representation.

Together, these stages move towards a unified framework for video representation learning that is grounded in language, enriched by spatial and motion priors, and general enough to support zero-shot transfer across diverse tasks—from classification and question answering to real-world interaction. Through extensive experiments, we demonstrate how aligning video learning with language not only improves interpretability and generalization, but also brings us a step closer to human-level internal models of the world.

1.2 Organization

The remainder of this proposal is organized as follows. We begin with an overview of related work, followed by four core chapters corresponding to the key stages of our approach: self-supervised learning with language guidance, spatial reasoning in multimodal models, long video understanding, and structured motion representation. We conclude with a discussion of future directions.

Chapter 2: Related Work. This chapter surveys prior literature relevant to our contributions. We discuss foundational approaches in self-supervised video learning, vision-language modeling, spatial localization in multimodal systems, motion understanding, and their applications in both recognition and control tasks. This contextualizes our proposed framework within the broader research landscape.

Chapter 3: Language-Guided Self-Supervised Learning (LSS). We introduce LSS, a self-supervised video learning framework that aligns video representations with language-derived concept spaces. By leveraging pre-trained image-language models like CLIP and extending them to the video domain, LSS achieves strong zero-shot performance without requiring per-video labels or captions.

Chapter 4: Enhancing Spatial Reasoning (LocVLM). This chapter presents our approach to improving spatial understanding in visual-language models through

language-based supervision of object locations. We demonstrate how representing coordinates as text and integrating them into instruction-tuning improves localization, spatial reasoning, and robustness to hallucination.

Chapter 5: Motion for Long Video Understanding (MVU). We extend our framework to long video understanding by constructing object-centric information streams—such as object trajectories—represented in natural language. This allows any off-the-shelf LLM to explicitly reason about motion and long-term dependencies, leading to improved performance on temporally complex tasks.

Chapter 6: Structured Motion Representations (LangToMo). We propose a structured representation of motion that captures fine-grained dynamics tied to language and visual state. By learning mappings from natural language to motion representations using large-scale video-caption data, LangToMo enables generalization extending even to real-world robotic control tasks.

Chapter 7: Conclusion and Future Work. We summarize the key contributions of the thesis and outline promising directions for future work. In particular, we propose to extend our framework to model motion in 3D and enable learning from videos that include ego motion. These capabilities would allow more general and physically grounded video representations, supporting richer understanding and interaction in real-world environments.

Chapter 2

Literature Review

2.1 Language Based Video Self-Supervised Learning

Self-Supervised Learning in Videos was initially dominated by pretext tasks specific to the video domain [3, 75, 99, 164, 169, 187, 189, 236, 252–254, 263]. Recently a shift to contrastive losses led to [55, 64, 82, 84, 190, 210] with some variants focused on video specific view generation [17, 38, 51, 95, 201]. An alternate direction has been masked auto-encoders [247]. To the best of our knowledge, existing video self-supervised learning (SSL) approaches operate purely within the visual domain. By video SSL, we refer to methods that utilize only videos with no paired captions (or labels) for each video during training. In contrast, our proposed LSS learns purely from videos in a self-supervised manner, integrating pre-trained language-image models to learn language aligned representations.

Zero-shot Action Recognition began with manual attribute and feature selection [68, 70, 100, 151, 308] with later works utilizing action word embeddings [26, 287]. The idea of connecting action categories with elaborate descriptions of those actions, within language embedding spaces [39, 330] has been a next step and is closely related to our work. This idea is also explored in image domain to boost zero-shot performance [165]. While our work is inspired by such approaches, in contrast, we use relations between such actions and descriptions as self-supervised signals for learning. Recent image CLIP models [103, 196] are another line of works achieving strong performance on some video classification tasks, with only single frame processing. Multiple approaches build on image CLIP [196] to learn video level representations [11, 108, 148, 158, 175, 257] under fully-supervised settings. While achieving strong performance on the training datasets, their zero-shot improvements over CLIP are minimal or even subpar (see Table 3.2). Therein, LSS

focuses on zero-shot performance under self-supervised settings while retaining (and improving) the generality of the representation space.

Self-training methods leverage pseudo-labels on unlabeled data [20, 232, 242] for supervised-fashion training. Recently they have been combined with CLIP models for zero-shot operation [107, 130]. While inspired by such self-training approaches, our proposed LSS differs in its continuous feature space self-distillation, language-based relations enforcing, video domain operation, and cross-dataset transfer for zero-shot operation.

Adapting image-CLIP models to video under fully-supervised settings has gathered much interest [43, 106, 192, 208, 288, 290]. Expanding backbones for temporal modeling, multi-modal fusion, secondary training objectives, partial parameter updates, and scaling-up data are key ideas explored [43, 108]. In contrast, LSS is a first to operate under self-supervised settings using no video annotations.

Contemporary work in [146] adapts image CLIP features to video tasks label free similar to our work. ViFi-CLIP [207] introduces zero-shot action recognition benchmarks and similarly adapts CLIP to videos retaining generality. Using LLMs for action recognition is also explored in [86].

2.2 Spatial Reasoning in Visual-LLMs

Localization in Contrastive Vision Language Models: Foundation vision language models (VLMs) such as CLIP [197] resulted in extensive exploration into language-tied localization in images both under dense (pixel / bounding-box) supervision [58, 60, 72, 79, 111, 129, 131, 137, 312, 315] and weak supervision [49, 157, 172, 203, 284, 285, 294, 321, 329]. Recovering explicit localization information within model representations has enabled more robust operation for certain tasks [203]. While our work differs from this contrastive setting given our use of LLM based generative predictions, we similarly explore how explicit location information within the language modality can improve V-LLMs.

Visual Large Language Models (V-LLMs): The advent of powerful large language models (LLMs) such as GPT-3 [29], Chat-GPT [181], and PaLM [47], as well as their open-source counterparts BLOOM [221], Vicuna [45], and LLaMA [248, 249], has resulted in direct use of these LLMs for computer vision tasks [81, 240]. Alternate lines of work explore how LLMs can be connected to existing visual foundation models [5, 9, 132, 150, 171, 200], in particular to CLIP visual backbones [197]. While earlier models explored large-scale (millions to billions of samples)

Method	Kosmos [188]	Ferret [297]	Shikra [37]	Proposed
Unified Arch.	✗	✗	✓	✓
Purely Textual	✗	✗	✓	✓
Pseudo Data	✗	✗	✗	✓
Video Domain	✗	✗	✗	✓

Table 2.1: Related Work Comparison: A unified architecture, purely textual inputs, pseudo data for scalable learning, and video domain operation distinguishes our proposed work from these prior methods.

image-text training [5, 9], later models [132, 150, 171] scale down on data dependency. LLaVA [150] in particular scales down on pre-training data to under 1 million image-text pairs, and use instruction fine-tuning [270] to enable human-style conversation with visual awareness. This is extended to video domain in [171, 205]. A shortcoming of these models is their lack of spatial awareness or location understanding in image space [37, 46, 74]. Spatial reasoning limitations in generative VLMs are studied in [46, 74]. Similar failures in captioning (and VQA) models are explored in [112]. A solution in [91] proposes code-generation based reasoning. Our work tackles these same limitations but follows an alternate direction of spatial-aware instruction fine-tuning. Another line of recent works [124, 188, 262, 297, 306, 320, 324] tackle this by introducing architectural modifications to explicitly extract region level features that are injected to the LLM as special tokens. While introducing extra tokens and layers, this also separates the localization task from language. In contrast, we use a generic architectures with purely textual location information (i.e. image space coordinates as text). Concurrent work in [37] explores this same idea, but we differ in 3 ways with, a) focus on optimal coordinate representation forms, b) data-efficient pseudo-labelling strategies, and c) video domain operation (see also Table 2.1).

Location Representations: Selecting regions within an image has a rich history in computer vision [161, 250] with greater focus on location outputs since the popularity of object detection [31, 40, 41, 73, 211, 246, 262]. Early anchor-based methods regress locations from anchor centers [73, 211], followed by direct location regression from object-level features [31, 246]. Recent works explore generative location predictions with diffusion processes [40] and sequence-generation [41, 262]. Ours resembles the latter given our use of an LLM, next token prediction objective, and sequential generation of textual location representations. However, [41, 262] utilize 1000 specialized location tokens (introduced to the LLM vocabulary) corresponding to 1000 bins uniformly spread across image space. While we explore

similar binning strategies, in contrast we introduce no additional tokens, focus on purely textual representation of locations, and explore multiple textual location representation forms.

2.3 Multimodal Language Models for Long Videos

Video Modality Exploration: Multiple recent works dissect the video modality into individual components [30, 198, 202, 305]. Single frame baselines are one interesting sub-class [21, 30, 52, 219, 325]. Extracting object-centric video modalities is another idea, spanning back to [52] which extracts multiple small objects from frames followed by modeling relations across frames and objects. Similarly, [219, 325] combine spatial information with single images to perform video tasks. However, these prior approaches focus on simple video tasks (i.e. action recognition) limited to visual modality. In contrast, our approach tackles the more complex language-tied task of long-video question answering that necessitates strong causal and temporal reasoning over long temporal windows. This task is also explored in [30], but we differ with likelihood selection, multi-modal information fusion, and usage of modern LLMs.

Long Video Question Answering: Long-video question-answering benchmarks are constructed to specifically test strong causal and temporal reasoning [276] over long temporal windows [162]. Early works explore querying objects or events based on referential and spatial relations [281, 303, 311], followed by focus on temporal modeling of sequential events [90, 127, 128, 277, 278]. While motivated by these works, MVU integrates such object information with large language models (LLMs) in a zero-shot manner requiring no video-level training. More recent works leverage LLMs [13, 184, 256, 268, 300] to directly perform these tasks but require video-caption training. In contrast, our MVU operates zero-shot on these tasks requiring no video-level training. Zero-shot operation is explored in [167, 261, 267, 269, 314], but we differ in using object-centric information modalities and efficient LLM sampling.

Large Language Model Reasoning: Recent LLMs [45, 47, 181] demonstrate multiple forms of strong reasoning abilities [48, 123, 154] including combining different information [271]. Their recent open-source variants [105, 243, 249] achieve equally promising skills using scaled-down models [105] while also demonstrating strong world knowledge [6, 136, 265, 283, 298, 326] even in domains such as robotics [138]. In our work, we leverage these strengths of LLMs for complex video-language tasks, focused on disentangling the effect of their abilities for video

QnA tasks.

Language based Fusion: The idea of fusing different modality information using natural language as a medium has been explored in multiple recent works [85, 86, 145, 200, 258, 310]. In [145, 200], language is utilized as an implicit medium for self-supervising video action recognition. Multimodal information represented as language is fused with visual information for action recognition and robotics tasks in [85, 86, 138, 258]. We utilize a similar language-as-a-medium fusion of multimodal information, but explore this in the context of complex video-language tasks. [310] is most similar to our work, but we differ with focus on long-video tasks and object-centric information.

2.4 Learning Motion Representations from Videos

Learning from Videos: Robot learning has a rich history of leveraging videos to extract sub-goal information, learn strong representations, or build dynamics models for planning [10, 36, 63, 66, 93, 118, 122, 125, 173, 185, 213, 224, 225, 231, 238, 239]. Several recent works learn representations connected to language modality from video-caption data [63, 93, 118, 238], but depend on additional action-trajectory annotations, pretrained segmentation models, or task-specific heuristics for robot control. We explore a similar direction, learning language-conditioned motion representations from video-caption data. In contrast to these works, our LangToMo learns representations that are *interpretable* and *motion-focused*, which we use for robot control with no additional supervision. Our focus on pixel motion also allows faster learning of more generalizable representations.

Pixel Motion to Actions: Robot navigation and control, especially in the context of aerial drones, has long benefited from optical flow representations [7, 53, 94, 126], inspired by animal perception system that use optical flow for stable control and movement [8, 12, 76, 217]. Video self-supervised learning has also extensively leveraged optical flow to learn motion representations [83, 226]. In contrast to prior works, our LangToMo is the first to model optical flow from a single image (pixel motion) conditioned on textual action descriptions, allowing language conditioned robot control.

Diffusion-Based Motion Generation: Diffusion models have emerged as powerful generative frameworks capable of capturing complex data distributions through iterative denoising processes [42, 44, 62, 71, 88, 89, 101, 121, 153, 199, 214, 229, 230, 251, 255, 317, 318]. While some works directly predict optical flow from

image pairs [156, 220], these tackle well-defined inputs. In contrast, LangToMo generates pixel motion from a single image and language command, capturing the multimodal nature of future motions. By also conditioning on past motion, our approach introduces temporal grounding, making it well-suited for robot control.

Language-Conditioned Robotic Manipulation: Several recent works use vision-language models for robot control [27, 28, 61, 93, 102, 114, 118, 139, 176, 178, 182, 212, 238, 245, 272, 304, 307, 328] taking advantage of large-scale training with web-scale vision-language data. In contrast to prior work using sequential language models, we learn motion representations under weak supervision (only video-caption data) using zero action trajectory annotations. We also utilize an image diffusion model similar to [93, 118, 238] but differ by learning universal and interpretable motion representations directly, which even allows conversion to robot actions directly with no further training.

Chapter 3

Language Based Video Self-Supervised Learning

3.1 Introduction

Actions in videos are defined by individual objects, their relationships, and interaction [2, 218]. Video self-supervised learning focuses on discovering representations aware of such action attributes directly from video content with no human supervision [222]. Particularly in the case of videos, where manual human annotation can be both expensive and noisy, such self-supervised approaches are invaluable.

A recent variant of self-supervision explores learning with loosely paired image-caption pairs, leading to highly transferable and robust representations such as CLIP [196]. These approaches obtain zero-shot performance often comparable to fully-supervised methods. However, their counterparts in the video domain [43, 106, 192, 208, 257, 288, 290] do not exhibit the same generality. In fact, some approaches training CLIP on videos [175, 257] perform subpar to image-CLIP under zero-shot settings (see Table 3.2). Such behaviour can be attributed to lesser availability and more noisy nature of labelled (or paired caption) video datasets [222]. This motivates exploration into self-supervised learning (SSL) techniques that can learn from videos under less supervision while utilizing existing image CLIP [196] like representations. Existing state-of-the-art video SSL approaches [201, 247] learn highly transferable representations from videos, but combining these with image CLIP representations is not straightforward. In fact, despite methods like SVT [201] being able to utilize image SSL representations [32] for weight initialization to achieve better performance, using image CLIP representations instead for weight initialization leads to performance subpar to image CLIP

(see Table 3.4). This raises necessity for alternate video SSL approaches compatible with CLIP like image representations and is our key motivation.

In this work, we explore self-supervised learning techniques that adapt image CLIP models [196] to video domain under entirely self-supervised settings, dependent on no form of video level labels or captions. Under this setting, natural language can still provide strong cues regarding attributes that compose an action category [26, 287]. We leverage this idea to propose a novel *language-based* self-supervised learning objective. Following a standard self-distillation and multi-view based SSL formulation [32, 201], we introduce language aligned feature spaces, *action concept spaces*, where our SSL objectives operate. Large-language models (LLMs) [29], given their extensive world knowledge [174, 322], serve as an ideal tool to generate necessary textual concepts for these spaces. We also introduce regularization suitable for our language aligned SSL objective to prevent collapse during training. Our resulting framework is termed *Language-based Self-Supervision*, or LSS.

In contrast to existing video self-supervised learning approaches [201, 247], our proposed LSS retains and improves transferability of image CLIP representations much better (see Tables 3.1 and 3.4). Additionally, our language aligned learning framework allows direct zero-shot operation on downstream tasks. Moreover, unlike video CLIP methods with similar zero-shot capabilities [43, 106, 192, 208, 257, 288, 290] that utilize per-video labels / captions for learning, our proposed LSS requires only videos for training.

We summarize our key contributions as follows:

- Self-supervised learning paradigm capable of retaining and improving strengths of CLIP like image representations for video domain operation
- Video specific self-supervised learning objectives, namely *concept distillation* and *concept alignment*, that enforce relations between action categories and their visual attributes
- Novel language-based video self-supervised learning framework operating zero-shot on downstream action classification tasks without requiring per-video labels / captions for training

Experiments on action recognition datasets showcase state-of-the-art performance for our learned representations under linear-probing, standard zero-shot, and transductive zero-shot settings.

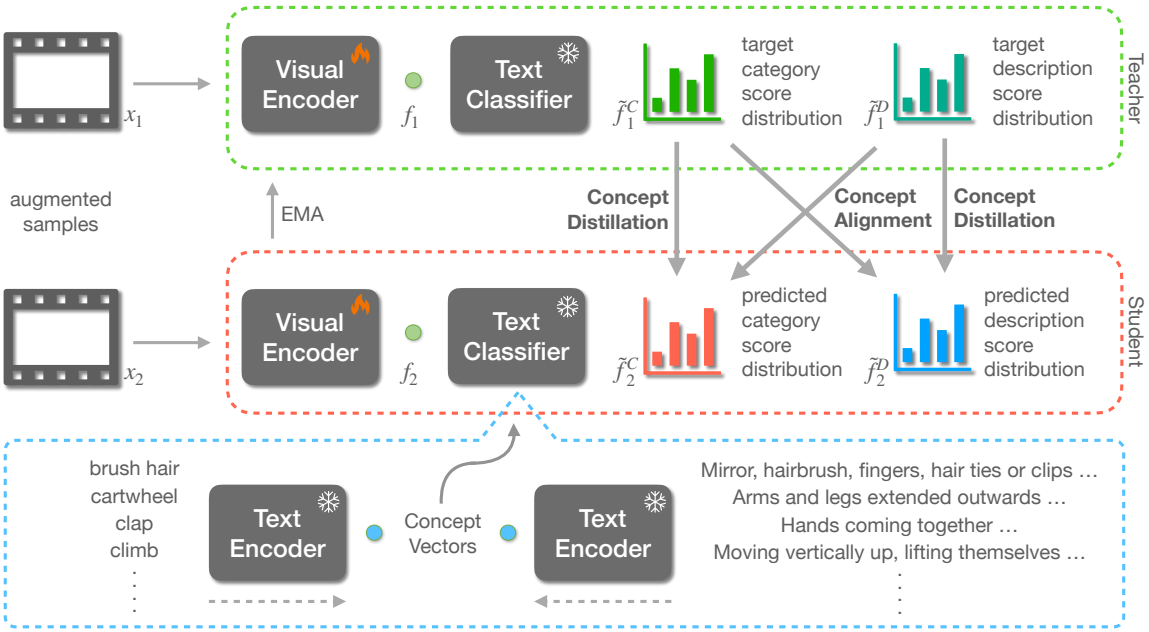

Figure 3.1: **Architecture Overview:** Our overall setup contains three components: visual teacher model (green), visual student model (red), and language model (blue). We utilize the text encoder of CLIP as our language model and extract *concept vectors* relevant to action labels and descriptions of those actions. A visual encoder (containing a space-time backbone) is partially initialized with CLIP's visual encoder and used to obtain sample specific features. Generated concept vectors are used to project these features to a *concept space* where our proposed *concept distillation* and *concept alignment* losses are applied.

3.2 Language-based Self-Supervision (LSS)

In this section, we present our proposal, Language-based Self-Supervision (LSS). The generality and robustness of shared image-language representation spaces such as that of CLIP [196] allow interesting manipulations of visual representations using language. We explore such manipulations under the setting of visual self-supervised learning focusing on video understanding. Self-supervised objectives can operate within a latent space constructed with language, retaining language alignment of learned visual representations. This allows better interpretability of representations as well as zero-shot inference. We discuss the four key components of our approach: backbone architecture, concept distillation objective, modifications to avoid collapse, and concept alignment objective.

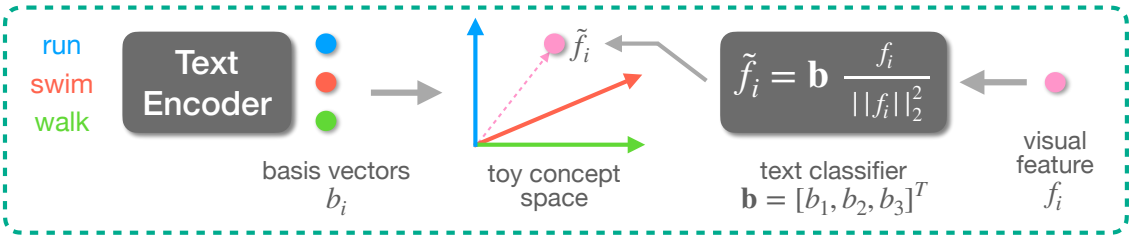

Figure 3.2: **Concept Spaces:** We illustrate a toy concept space constructed with the three action concepts: run, swim, and walk. In this example, the text classifier projects visual feature f_i into the 3-dimensional toy concept space to produce $\tilde{f}_i$.

3.2.1 Backbone Architecture

Our approach introduces a *text classifier* to self-distillation based SSL works [32, 201], in place of the projector network. Given a data sample x, let $x_1, x_2 \in \mathbb{R}^{(C,T,H,W)}$ be two augmented views generated using video specific transformations following [201], where $C = 3, T = 8, H = W = 224$ are channel, time, and spatial dimensions respectively.

Visual Encoder: A visual encoder, θ_v, processes x_i to produce feature $f_i \in \mathbb{R}^{768}$. We utilize the pre-trained image encoder of CLIP [196] expanded for temporal modelling using factorized space-time attention. The vision transformer variant of CLIP is selected to allow our factorized space-time attention. In particular, we use ViT-B/16 architecture for the the image encoder, in which for a given augmented view with $H = W = 224$ and $T = 8$, each transformer block processes 8 temporal and 196 spatial tokens separately in sequential order, and the embedding dimension of each token is $\mathbb{R}^{768}$. In addition to the input tokens from the data sample, one classification token [54, 59] serves as the final feature vector output by the network, namely f_i, which is common to the CLIP image encoder. This classification token is inflated and processed suitably following [19] to accommodate our modifications for factorized space-time attention. We follow [19] to zero-initialize additional time-attention parameters, achieving outputs identical to the pre-trained CLIP image encoder at start of training.

Text Classifier: Inspired by [274], a set of n language embeddings extracted from the CLIP text encoder, θ_t, are used to construct the weight parameter of a linear layer (with no bias term), which we call our text classifier, θ_c. The role of this text classifier is to project visual features f_i to a vector space defined by those n embeddings, producing $\tilde{f}_i \in \mathbb{R}^n$. Next we discuss these vector spaces (referred to as action concept spaces) and the text classifier module in detail.

3.2.2 Action Concept Spaces

Self-supervised learning approaches following exponential moving average (EMA) based self-distillation [32, 78, 201] utilize a projector network (MLP) to operate in a higher dimensional feature space. This is expected to minimize train-test domain gaps, handle noisy positive sample pairs, and better discriminate nuanced feature differences [14]. Focused on these notions, we propose an alternate *concept space* composed of a set of basis vectors defined by language-based action concepts. Our language-based self-supervision objectives operate within such concept spaces.

Concept Spaces: Building off the assumption that text encoder features capture subtle differences between distinct actions categories, we hypothesize that necessary nuanced distinctions between these actions will be better captured in our proposed concept spaces. The defining parameters of concept spaces are their basis vectors, b_i. Normalized embeddings (extracted from text encoder, θ_t) of various natural language captions (c_i) relevant to action categories are used as these basic vectors.

$$b_i = \theta_t(c_i) \; / \; \|\theta_t(c_i)\|_2^2 \tag{3.1}$$

$$\mathbf{b} = [b_1, b_2, ... \; b_n]^T \; ; \mathbf{b} \in \mathbb{R}^{(n,d)} \tag{3.2}$$

Note that these basis vectors are not necessarily orthogonal. As illustrated in Fig. 3.2, a single set of basis vectors, $\mathbf{b}$, defines one action concept space. We define two sets of basic vectors: action category vectors and action description vectors. Action category vectors relate to a single action label which is converted to a caption using textual prompting following [196]. Action description vectors are averaged embeddings of multiple descriptions and visual characteristics relevant to individual action categories. These two distinct sets of basic vectors lead to two distinct concept spaces which we name *category concept space* and *description concept space* respectively.

Category Concept Space: We explore 3 different strategies to construct the category concept space. The base setup uses action labels from Kinetics-400 [33], UCF-101 [234], and HMDB-51 [120] datasets, leading to a set of 530 (400 + 101 + 51, ignoring overlaps) basis vectors. Our next goal of connecting LLMs and their action awareness occurs in the second two strategies. We utilize LLMs [29] and visual-LLMs [149] to extract large sets of action category labels. While we explore this idea of expanding the basis vector set with LMM based additional action labels in Section 3.3, the base setup containing a modest 530 categories was sufficient to improve downstream task performance.

Description Concept Space: This space is constructed conditioned on the

previous category concept space. For each action label used in the latter, we extract 4 distinct descriptions and a set of visual characteristics relevant to that action label using a large language model (LLM). The role of the LLM is to inject its world knowledge (i.e. awareness on videos, actions, and their attributes) into our learned representations during self-supervised learning. In detail, we prompt GPT-3 [29] to generate such descriptions and characteristics using procedure outlined in Section 3.5.1. We highlight that GPT-3 is used here as an intelligent LLM containing world knowledge on videos and actions, in order to create natural language descriptions for given action category labels. The textual outputs generated for each action label are processed by our text encoder to produce multiple embeddings for a single action label. These embeddings are averaged to produce the corresponding basis vector for the description concept space. Note how this leads to a common dimensionality between the two concept spaces as well as one to one correspondences between the basic vectors of the spaces, which we leverage in our self-supervision objectives.

3.2.3 Concept Distillation

We now describe our primary self-supervised learning objective, concept distillation. Standard multi-view based self-supervision enforces a network to encode the common information between two augmented (distorted) views of a data sample [14]. This common information can be considered as the augmentation invariant signal present in the original data sample [14, 16]. In the case of self-distillation based approaches [32, 201], a higher dimensional feature space is utilized to enforce the self-supervision objectives. Instead, we propose to use action concept spaces as an alternative.

Proposed concept distillation depends on an action concept space and visual video features aligned to the basis vectors of that space. Given our visual features $f_i \in \mathbb{R}^d$, we obtain projected $\tilde{f}_i \in \mathbb{R}^n$ as,

$$\tilde{f}_i = \mathbf{b}\ (f_i / \|f_i\|_2^2) = [b_1 \cdot f_i', b_2 \cdot f_i', \ldots\ b_n \cdot f_i']^T \tag{3.3}$$

Similarity Calculation: Projecting normalized visual video features to a concept space corresponds to calculating the dot-product similarity with each basic vector of the concept space. The projected vector $\tilde{f}_i$ can be viewed as a similarity *score distribution* across all basis vectors of the concept space. Inspired by [274], we implement this similarity calculation as a linear layer with weight matrix $\mathbf{b}$ and bias terms zero. We refer to this layer as the *text classifier*. Similar to [274], our text

classifier remains frozen (no parameter updates), but in our case, this is to retain the original language distribution.

Concept Distillation Objective: Viewing projected features for two augmented views of a single video as score distributions, we argue that the underlying signal of the original video would relate to a unique score distribution to which score distributions of each view should be similar. Therein, following our EMA teacher based self-distillation setup (see Section 3.2.1 for details), we enforce the score distribution to be consistent across views. Given two views x_1, x_2 of a single video, our teacher and student visual encoders process them respectively to produce f_1, f_2. The text classifier projects these to concept space, producing score distributions $\tilde{f}_1, \tilde{f}_2$. We obtain our objective, $\mathcal{L}_{\text{CD}}$ as:

$$\hat{f}_i[k] = \frac{\exp(\tilde{f}_i[k]/\lambda_i)}{\sum_{j=1}^{n} \exp(\tilde{f}_i[j]/\lambda_i)} \tag{3.4}$$

$$w_s = \max(\hat{f}_1) \tag{3.5}$$

$$\mathcal{L}_{\text{CD}}(\tilde{f}_1, \tilde{f}_2) = -w_s \cdot \sum_{j=1}^{n} \hat{f}_1[j] \log \hat{f}_2[j] \tag{3.6}$$

The teacher and student score distributions, $\tilde{f}_1, \tilde{f}_2$, are softmax normalized in Eq. (3.4), with temperature terms $\lambda_1 = 0.1, \lambda_2 = 1$ for sharpening only the teacher score distribution. A significance score w_s is calculated for each sample in Eq. (3.5). In the softmax normalized teacher score distribution ($\hat{f}_1$), the maximum value is high when peaked at a single action concept and low when peaked at multiple action concepts. Considering the noisy nature of multi-peak teacher score distributions, we utilize w_s to minimize their overall effect during training. Our overall $\mathcal{L}_{\text{CD}}$ is thus implemented as in Eq. (3.6).

Distinct Concept Spaces: Given the two distinct action concept spaces defined in Section 3.2.2, we utilize two parallel text classifiers to implement each, and obtain two score distributions, one for each concept space. Defining score distributions $\tilde{f}_i^C, \tilde{f}_i^D$ for category and description concept spaces respectively, we apply our $\mathcal{L}_{\text{CD}}$ on each pair separately to obtain two losses $\mathcal{L}_{\text{CD}}^{\text{X}}$ for X∈{C,D} as:

$$\mathcal{L}_{\text{CD}}^{\text{X}} = \mathcal{L}_{\text{CD}}(\tilde{f}_1^{\text{X}}, \tilde{f}_2^{\text{X}}) \tag{3.7}$$

We highlight how our concept spaces implemented as text classifiers are maintained intact by freezing the text classifier during training. This allows our approach to perform direct zero-shot inference, making concept distillation additionally advantageous over standard video SSL techniques.

3.2.4 Uniform Distribution Prior

Avoiding collapse is a key concern in SSL methods [14, 32, 201] and recent self-distillation based approaches utilize feature sharpening and centering operations to avoid collapse [32, 201]. While we similarly perform sharpening operations on the teacher outputs, given the nature of our action concept space, performing a learned vector mean subtraction based centering operations can break the meaningful structure of score distributions. Instead, we enforce a uniform distribution prior on the expected score distribution over the entire training dataset. The centering operation proposed in [32] acts similarly pushing representations towards a uniform distribution while the sharpening operation counters its effect. We approximate expectation over the dataset as a moving average of mean score distributions at each train iteration and the uniform prior is enforced as:

$$\hat{f}^{\mathrm{X}}_{\mathrm{MA}} = \tau \cdot \hat{f}^{\mathrm{X}}_{2} + (1-\tau) \cdot \hat{f}^{\mathrm{X}}_{\mathrm{MA}} \tag{3.8}$$

$$\mathcal{L}^{\mathrm{X}}_{\mathrm{UP}} = -\frac{1}{n} \sum_{j} \log \hat{f}^{\mathrm{X}}_{\mathrm{MA}}[j] \tag{3.9}$$

where the hyper-parameter $\tau = 0.5$ is fixed during training. We highlight that $\mathcal{L}_{\mathrm{UP}}$ is necessary for convergence with concept distillation and is added to the concept distillation objective, $\mathcal{L}^{\mathrm{X}}_{\mathrm{CD}}$.

3.2.5 Concept Alignment

Aligning action category labels and their descriptions or attributes within some embedding space has been explored in video SSL under multiple settings [39, 330]. Motivated by these promising results, we explore how such alignment can be integrated to improve our framework with *concept spaces*. In Section 3.2.2, we define two distinct action concept spaces constructed from category labels and detailed category descriptions respectively. We hypothesize that explicit alignment of video features between these two spaces based on their one to one relationship can learn additional information. Therein, we introduce our concept alignment objective, $\mathcal{L}_{\mathrm{CA}}$, as follows:

$$\mathcal{L}_{\mathrm{CA}} = \mathcal{L}_{\mathrm{CD}}(\tilde{f}^{\mathrm{C}}_{1}, \tilde{f}^{\mathrm{D}}_{2}) + \mathcal{L}_{\mathrm{CD}}(\tilde{f}^{\mathrm{D}}_{1}, \tilde{f}^{\mathrm{C}}_{2}) \tag{3.10}$$

Overall SSL Objective: Reusing $\mathcal{L}_{\mathrm{CD}}$ from Eq. (3.6), we match score distributions across our two concept spaces instead of within a single concept space. $\mathcal{L}_{\mathrm{CD}}(\tilde{f}^{\mathrm{C}}_{1}, \tilde{f}^{\mathrm{D}}_{2})$

aligns student description score distribution $\tilde{f}_2^{\mathrm{D}}$ to teacher category score distribution $\tilde{f}_1^{\mathrm{C}}$ while $\mathcal{L}_{\mathrm{CD}}(\tilde{f}_1^{\mathrm{C}}, \tilde{f}_2^{\mathrm{D}})$ aligns student category score distribution $\tilde{f}_2^{\mathrm{C}}$ to teacher description score distribution $\tilde{f}_1^{\mathrm{D}}$. Combining all terms, we obtain:

$$\mathcal{L} = (\mathcal{L}_{\mathrm{CD}}^{\mathrm{C}} + \mathcal{L}_{\mathrm{UP}}^{\mathrm{C}}) + (\mathcal{L}_{\mathrm{CD}}^{\mathrm{D}} + \mathcal{L}_{\mathrm{UP}}^{\mathrm{D}}) + \mathcal{L}_{\mathrm{CA}} \tag{3.11}$$

3.2.6 Concept Space Variants

Our baseline concept space (described in Section 3.2.2) utilizes labels from three standard video datasets (Kinetics-400, UCF-101, HMDB-51). However, we want to ensure scalability with more data and no label leakage to downstream evaluation tasks. With this goal, we propose 2 additional variants of action concept spaces tagged LSS-B and LSS-C. These variants do not use any form of ground truth textual labels from datasets. Moreover, they leverage the world awareness (i.e. knowledge on videos and actions) of LLMs to generate extensive action categories. Our baseline setup is hereafter referred as LSS-A.

For LSS-B, we use GPT-3 [29] to generate a large set of action labels. We first prompt GPT to categorize all common human actions / activities into 20 groups. For each group, we again ask GPT to generate at least 100 visually diverse action categories. These are all collected to create a set of 2000 action labels. We then use projections of these labels in CLIP text-encoder representation space to eliminate labels of high semantic similarity (spectral clustering in feature space from [204] to identify similar features), achieving 1000 diverse action categories. So our 1000 action categories for LSS-B are generic, not tied to any of our training datasets, and scalable with more data.

For LSS-C, we generate a label set using only videos from the training dataset. We use PCA based clustering to identify 2000 representative videos from a randomly sampled subset (50,000) of our training dataset and then use image-captioning models (LLaVa [149]) on video center frames to generate a diverse set of 2000 action labels. This is further reduced to 500 eliminating labels that are similar in feature space of the CLIP text encoder. In this case, our generated labels are tied to the training dataset, but uses no textually annotated category labels. We use only the videos (and an image-to-text captioning model) to generate our label set, still resulting in a scalable framework.

Note that each of these alternate strategies relates to construction of our category concept space. Given the selected set of textual category labels of this space, the description concept space is constructed in the same common way (as described in Section 3.2.2). We also reiterate that LSS-B and LSS-C variants use no category information from train / test datasets.

3.3 Experiments

In this section, we first describe our experimental setup followed by discussion of results for linear probing self-supervised representations and zero-shot analysis.

Datasets: We use three standard action recognition benchmark datasets in our experiments: Kinetics-400 [33], UCF-101 [234], and HMBD-51 [120]. Kinetics-400 is a large-scale dataset containing 240,000 training videos and 20,000 validation videos belonging to 400 different action classes. On average, these videos are of duration around 10 seconds, with 25 frames per second (i.e., around 250 frames per video). UCF-101 and HMBD-51 are small-scale datasets each containing 13k videos (9.5k/3.7k train/test) belonging to 101 classes and 5k (3.5k/1.5k train/test) videos belonging to 51 classes respectively. They also contain similar duration videos.

Self-supervised Training: Our SSL training phase uses the train split of Kinetics-400 dataset [33] *without* using any per-video labels. We train for 15 epochs using a batch size of 32 across 4 NVIDIA-A5000 GPUs using ADAM-W [116, 155] optimizer on the student model with an initial learning rate of $1e-5$ following a cosine decay schedule. The EMA teacher is updated from student weights after each training iteration with a decay ratio of $2e-4$. Unless otherwise specified, this model is used for all downstream task evaluations.

Transductive Training: For selected experiments, we additionally perform self-supervised training directly on the train split of each downstream dataset. For Kinetics-400, we follow the same setup described above. In the case of HMDB-51 and UCF-101, we perform self-supervised training for a longer duration of 100 epochs (smaller train sets) leaving all other hyper-parameters unchanged.

View Generation: Our self-supervised setup requires two views of a single video. We sample two clips from a video following global view generation in [201]. In detail, we select two random intervals from a video, and uniformly sample (equal time gaps between frames) 8 frames of 224x224 spatial dimensions from within that interval. Standard video augmentations from [191] are also applied randomly for each view.

Linear Probing: We follow standard linear probing settings on our two downstream datasets to evaluate quality of representations learned by our self-supervised learning phase. We follow the same settings in [201] for fair comparison. Our visual encoder is frozen and a randomly initialized linear layer is trained on the train split of the downstream dataset in a fully-supervised manner. We train for 15 epochs using a batch size of 128 across 4 NVIDIA-A5000 GPUs using ADAM-W [116, 155] optimizer with an initial learning rate of $1e-3$ following a cosine decay

Table 3.1: **Linear Probing on HMDB-51 [120] and UCF-101 [235]:** We compare our method against prior work, reporting top-1 (%) accuracy (following evaluation procedure in [201]). 'ITP' stands for image-text pre-training. Gray shaded methods use additional optical flow (OF) inputs for training. Nevertheless, our performance is comparable to such methods using per-video OF modality information. In contrast, we use generic language modality information and require no one-to-one language relations with individual videos to train.

Method	Backbone	ITP	TFLOPS	Frames	Epochs	HMDB	UCF
MemDPC [82] (ECCV '20)	R2D3D-34	✗	-	64	-	30.5	54.1
CoCLR [84] (NeurIPS '20)	S3D	✗	0.07	32	100	52.4	77.8
VideoMoCo [183](CVPR '21)	R(2+1)D	✗	17.5	32	200	49.2	78.7
CVRL [190](CVPR '21)	R3D-50	✗	3.19	32	800	57.3	89.2
MoDist [275] (Arxiv '21)	R3D-50	✗	3.19	8	100	63.0	91.5
BraVe [210](ICCV '21)	R3D-50	✗	3.19	16	-	68.3	92.5
Vi^{2}CLR [57] (ICCV '21)	S3D	✗	0.07	32	300	47.3	75.4
MCN [147] (ICCV '21)	R3D	✗	3.19	32	50	42.9	73.1
CORP [92] (ICCV '21)	R3D-50	✗	3.19	16	800	58.7	90.2
SVT [201](CVPR '22)	ViT-B	✗	0.59	16	20	57.8	90.8
VideoMAE [247](NeurIPS '22)	ViT-B	✗	0.59	16	800	60.3	84.7
MERLOT [309](NeurIPS '21)	ViT-B	✓	-	16	-	55.4	80.1
VATT [4](NeurIPS '21)	ViT-B	✓	-	32	-	66.4	87.6
TVTS [313](CVPR '23)	ViT-B	✓	0.59	16	20	58.4	83.4
LaViLa [323](CVPR '23)	ViT-L	✓	-	4	5	61.5	88.1
LSS-A (ours)	ViT-B	✓	0.59	8	20	69.2	91.0
LSS-B (ours)	ViT-B	✓	0.59	8	20	**69.4**	**91.1**
LSS-C (ours)	ViT-B	✓	0.59	8	20	69.1	90.8

schedule. During inference, we sample three 224x224 dimensional spatial crops with 8 uniformly spaced frames from each video following prior work [190, 201].

Zero-Shot Inference: For zero-shot inference, we project class labels of downstream datasets to our text encoder feature space, and construct an alternate text classifier. Using this text classifier, we make zero-shot predictions. This setup is identical to dot-product similarity based inference in CLIP [196] (explanation in Section 3.2.3). In line with prior work [190, 201], we feed three 224x224 dimensional spatial crops with 8 uniformly spaced frames sampled from each video to the visual encoder and average its output feature embedding prior to normalized dot-product calculation in the text encoder.

Table 3.2: **Zero-shot Transfer on HMDB-51 [120] and UCF-101 [235]:** We compare LSS against prior work, reporting top-1 accuracy (%). Mean across three test splits is reported following [108]. 'ITP' stands for image-text pre-training and 'Video Labels' refers to using per-video annotations (or paired captions) for supervision during video-based training. We highlight how among directly comparable unsupervised (at video level) approaches as well as over the CLIP [196] baseline, LSS boosts zero-shot performance.

Method	Backbone	ITP	Video Labels	Frames	HMDB	UCF
TS-GCN [69]$_{\text{(AAAI '19)}}$	GCN	✗	✓	16	23.2	34.2
E2E [25]$_{\text{(CVPR '20)}}$	CNN	✗	✓	16	32.7	48.0
ER-ZSAR [39]$_{\text{(ICCV '21)}}$	CNN	✗	✓	8	35.3	51.8
ActionCLIP [257]	ViT-B	✓	✓	32	40.8	58.3
X-CLIP [175]$_{\text{(ECCV '22)}}$	ViT-B	✓	✓	32	44.6	72.0
VicTR [108]	ViT-B	✓	✓	32	51.0	72.4
ViFi [207]$_{\text{(CVPR '23)}}$	ViT-B	✓	✓	32	51.3	76.8
MTE [286]$_{\text{(ECCV '16)}}$	-	✗	✗	-	19.7	15.8
ASR [259]$_{\text{(ECML '17)}}$	CNN	✗	✗	16	21.8	24.4
ZSECOC [193]$_{\text{(CVPR '17)}}$	-	✗	✗	-	22.6	15.1
UR [330]$_{\text{(CVPR '18)}}$	CNN	✗	✗	1	24.4	17.5
CLIP [196]$_{\text{(ICML '21)}}$	ViT-B	✓	✗	1	46.5	69.8
CLIP [196]$_{\text{(ICML '21)}}$	ViT-B	✓	✗	8	47.2	70.3
LaViLa [323]$_{\text{(CVPR '23)}}$	ViT-L	✓	✗	4	16.6	18.2
LSS-A (ours)	ViT-B	✓	✗	8	49.5	72.0
LSS-B (ours)	ViT-B	✓	✗	8	50.2	73.8
LSS-C (ours)	ViT-B	✓	✗	8	**51.4**	**74.2**

3.3.1 Linear-Probing Analysis

We first evaluate LSS under linear probing settings on HMDB-51 & UCF-101 datasets. Our results (top-1 accuracy) are reported in Table 3.1. Our proposed LSS achieves state-of-the-art results on both datasets, outperforming prior approaches. Note that MoDist [275] and BraVe [209], both of which additionally utilize video-level optical flow (OF) for self-supervision, are not directly comparable. Still, our LSS showcases competitive performance to those, even without such motion information.

Table 3.3: **Transductive Zero-shot Transfer on HMDB-51 [120], UCF-101 [235], and Kinetics-400 [33]:** We report top-1 accuracy (%) following the evaluation procedure in [108]. Similar to prior work, we perform dataset specific unsupervised fine-tuning (using our self-supervised objective) on the train-splits of each downstream dataset (no labels used). 'ITP' refers to image-text pretaining, and 'Video Labels' refers to video level supervised training. Note that CLIP [196] is not transductive and is included only for comparison purposes.

Method	Backbone	ITP	Video Labels	Frames	HMDB	UCF	K400
UR [330]$_{\text{(CVPR '18)}}$	CNN	✗	✗	1	28.9	20.1	-
TS-GCN [69]$_{\text{(AAAI '19)}}$	GCN	✗	✓	16	23.2	34.2	-
CLIP [196]$_{\text{(ICML '21)}}$	ViT-B	✓	✗	8	47.2	70.3	49.7
MUST [130]$_{\text{(ICLR '23)}}$	ViT-B	✓	✗	1	48.9	**81.1**	51.2
LSS (ours)	ViT-B	✓	✗	8	**55.0**	75.6	**54.3**

3.3.2 Zero-Shot Analysis

Our LSS provides the additional advantage of zero-shot operation unlike standard video SSL approaches. To this end, we conduct two forms of zero-shot experiments. First, we evaluate LSS on standard zero-shot classification, where our model trained on Kinetics-400 (under SSL settings) is evaluated on the two downstream datasets, HMDB-51 and UCF-101. We report these results (top-1 accuracy) in Table 3.2. Compared to prior work utilizing per-video labels / captions for training, we achieve competitive performance. We note that MOV [192] trained under supervised settings with per-video labels and additional audio information is not a direct comparison.

In contrast to most prior approaches, LSS uses no video level labels for its Kinetics-400 training. In particular, LSS has not seen any labelled videos during its training process. Compared to prior work operating under these settings, LSS achieves state-of-the-art performance on both downstream datasets as seen in the bottom half of Table 3.2.

An alternate setting in prior zero-shot work is transductive training, where self-supervised learning is perfomed directly on train splits of downstream datasets. Under this setting, we evaluate on all three datasets, Kinetics-400, HMDB-51, and UCF-101, reporting results (top-1 accuracy) in Table 3.3. In the case of HMBD-51 and Kinetics-400, our method achieves state-of-the-art performance. For UCF-101, we achieve competitive results, and clear improvements over a CLIP [196] baseline.

Table 3.4: **Ablation on SSL objectives:** We ablate our proposed concept distillation (CD) applied on category (CD^C) and description (CD^D) concept spaces and concept alignment (CA) using linear probing (LP) & zero-shot transfer (ZS) on HMDB-51 [120] dataset. Since our approach cannot be trained without any objective, we construct two new baselines from CLIP [196] and SVT [201] (details in Section 3.3.3). Each proposed component obtains clear improvements over the baselines.

	CD^C	CD^D	CA	LP	ZS
CLIP	✗	✗	✗	63.9	46.5
CLIP†	✗	✗	✗	67.3	47.2
SVT§	✗	✗	✗	62.2	-
Ours	✓	✗	✗	68.5	48.8
Ours	✓	✓	✗	69.0	49.2
Ours	✓	✓	✓	69.2	49.5

Table 3.5: **Concept Space Ablation:** We report zero-shot accuracy (%) on HMDB-51 and UCF-101 datasets. 'K400 only' shows transfer to unseen downstream classes. The labels of K400 overlapping with UCF/HMDB are not used here. A CLIP [196] baseline (modified for video domain without retraining) is reported in row 1 for comparison purposes. For 2000 words & 10,000 sentences, we utilize most common nouns / verbs and example sentences for action verbs from WordNet [65, 166].

Method	Action Labels	HMDB	UCF
CLIP	-	47.2	70.3
Ours	K400 only (w/o U,H)	48.4	71.1
Ours	K400+U+H (A)	49.5	72.0
Ours	(A)+2000 words	48.6	71.8
Ours	(A)+10K sentences	49.6	71.4

3.3.3 Ablations

We next study the contribution of each component within our approach. All ablative experiments follow the same SSL phase on the Kinetics-400 train set (as described in Section 3.3) followed by zero-shot analysis on validation sets of HMDB-51 and UCF-101. In the case of linear probing results, training is conducted following same settings (see Section 3.3) on the train set of HMDB-51 followed by evaluation on its validation set.

SSL Objectives: First we ablate each proposed component in Eq. (3.11) and report results in Table 3.4. In addition to a direct CLIP [196] baseline, we construct two additional baselines building off CLIP [196] and SVT [201] for better comparison. CLIP† baseline applies our backbone modifications (for temporal modeling) with no training, which is identical to averaging per-frame visual encoder features. SVT§ baseline performs SVT [201] training with CLIP visual encoder initialization (note that language alignment breaks and zero-shot operation is not possible for this baseline). In comparison to the CLIP baselines, each proposed component, concept distillation in category and description concept spaces as well as concept alignment, leads to improvements. The comparison against SVT§ highlights how our SSL approach better preserves language aligned information (contained in

Table 3.6: **Ablation on Regularization and Significance Weight:** The effect of proposed regularization (left) and significance weight (right) is demonstrated. UDP regularization (see Section 3.2.4) is particularly essential to prevent collapse during the SSL training phase. Significance weight (w_s) also improves performance.

Method	HMDB	UCF
LSS	48.4	71.1
LSS w/o UDP	33.4	54.3

Method	HMDB	UCF
LSS	48.4	71.1
LSS w/o w_s	47.2	70.3

CLIP) that is useful even in linear probing. In contrast, the lower performance of SVT[§] compared to CLIP baselines indicates that generic SSL techniques may be losing useful information contained in CLIP.

Concept Spaces: Our next focus is on construction of concept spaces. We explore how separately augmenting each concept space affects downstream task performance measured with zero-shot transfer. These results are reported in Table 3.5. First, focused on the category concept space, we construct additional category labels using 1000 most common nouns and verbs each (total of 2000) from the WordNet dataset [65, 166]. Next, we augment the description concept space using 10,000 sentences. We select these from example sentences provided for action verbs in the WordNet dataset [65, 166]. In these experiments, only the concept distillation objective is applied on these augmented spaces and concept alignment operates only on the base category set. This is because independently augmenting one of the action spaces eliminates their shared and aligned dimensionality. Results for these two settings are reported in row 2 & 3 respectively in Table 3.5.

Uniform Distribution Prior: We ablate on proposed uniform distribution prior (UDP) which acts as a regularization to prevent collapse (Section 3.2.4). Our results in Table 3.6 (left) indicate clear necessity of such regularization to prevent collapse during SSL training.

Significance Weight in Concept Distillation: In Eq. (3.6), we utilize a significance weight term, w_s, which represents the confidence of the target concept space projection for a given sample. We note how each sample during training is a clip sampled from a video (which covers a temporal crop of video). Our intuition for this weight is to act as a way of prioritizing more important clips over the less important ones. Our ablations in Table 3.6 (right) indicate usefulness of this weight term.

3.4 Conclusion

We introduce a novel language-based self-supervised learning (SSL) approach for videos, termed LSS, capable of adapting strong language-aligned image representations (CLIP [196]) to the video domain. In particular, we propose two self-distillation based SSL objectives, *concept distillation* and *concept alignment*. Our approach trains with no video level labels or paired captions similar to prior video SSL works, but retains language alignment from image CLIP enabling direct zero-shot inference. We demonstrate state-of-the art performance in terms of linear probing with the learned representations on downstream tasks. For zero-shot operation, LSS demonstrates strong performance under both standard and transductive settings, indicating a promising direction for video SSL.

Limitations, Future Work, & Broader Impact: The language alignment of LSS may be limited mostly to per-frame static information since the alignment is derived from image CLIP [196]. LSS cannot distinguish motion based categories like "`moving object left to right`". Moreover, while containing highly discriminative and generic information at image level, CLIP features lack spatial awareness at an object level [204]. Our proposed model building off these representations in inherently limited in understanding object level motion and interaction within videos. However, recent progress in localization aware CLIP models [204, 284, 285] opens avenues for leveraging their object-centric or pixel-level representations to better model such video motion patterns, opening up interesting future directions. In terms of broader impact, the datasets and pre-trained models we use possibly contain biases, which may be reflected in LSS. However, our reduced reliance on human annotations may lower additional biases.

3.5 Additional Details

3.5.1 Prompting details

Our proposed approach utilizes two sets of language based captions: categories and descriptions. While categories are obtained directly from the class labels of datasets (set of unique labels - e.g. 400 classes in Kinetics-400 dataset), the descriptions are generated automatically utilizing GPT-3 [29]. For each category caption, we query GPT-3 to provide a set of descriptions and visual characteristics.

In detail, we use the following two prompts to generate descriptions and visual

Table 3.7: We report top-1 (%) accuracy on the Kinetics-400 [33] validation set for linear probing evaluation (left). All models are pre-trained on the training set of Kinetics-400 dataset. We also report a CLIP baseline for comparison purposes. Performance of our proposed approach is on-par with prior state-of-the-art and showcases improvements over our baseline method. We also report retrieval scores (top-right) for MSR-VTT and classification mAP (bottom-right) for Charades dataset.

Method	Backbone	Acc (%)
CVRL [190] (CVPR'21)	R3D-101	67.6
BraVe [210] (ICCV'21)	R3D-50	66.7
Vi^{2}CLR [57] (ICCV '21)	S3D	63.4
CORP [92] (ICCV '21)	R3D-50	66.6
SVT [201] (CVPR '22)	ViT-B	68.1
VideoMAE [247] (NeurIPS '22)	ViT-B	61.3
CLIP [196]	ViT-B	66.4
LSS (ours)	ViT-B	67.3

Method	R@1	R@5	R@10
CLIP [196]	30.6	54.4	64.3
LSS (ours)	33.8	58.2	70.3

Method	Classification mAP
CLIP [196]	19.7
LSS (ours)	23.1

characteristics:

```
prompt1 = "Give 4 different descriptions for the phrase: {category}?"
prompt2 = "List visual objects or characteristics usually seen with the action: {category}?"
```

The resulting two sets of captions are converted to text embeddings using our text-encoder, and a single average text embedding is computed. This averaged embedding is used as the description basis vector for that category. Also, the resulting dataset containing these category-description pairs is made available publicly.

3.5.2 Additional Experiments

Linear Probing Evaluation: We present more results for linear probing in Table 3.7 (left). Our proposed LSS improves over the baseline achieving competitive performance on Kinetics-400.

Text-to-video retrieval: An important characteristic of CLIP [196] is its retrieval ability across both language and visual modalities. In order to verify if proposed LSS retains these strengths, we run experiments on MSR-VTT text-to-video retrieval benchmark. We demonstrate how LSS improves over our baseline CLIP, reporting

these results in Table 3.7 (top-right).
Charades Evaluation: We explore an alternate task of zero-shot multi-label classification on the Charades video dataset. We report mAP results for this task in Table 3.7 (bottom-right) as an additional point of comparison.

Chapter 4

Learning to Localize Objects Improves Spatial Reasoning in Visual-LLMs

4.1 Introduction

Holistic visual understanding requires learning beyond simply content of an image to encompass awareness on spatial locations of objects and their relations [163]. In the context of visual question answering (VQA), such spatial awareness allows better reasoning involving structural and contextual information contained within an image [37].

Since the introduction of powerful large-language models (LLMs) such as GPT-3 [29], Chat-GPT [181], Vicuna [45], and LLaMA [248, 249] that are capable of human style conversation, their visual counterparts such as BLIP-2 [132], LLaVA [150] have enabled novel tasks within the vision modality. However, despite their [132, 150] highly generic visual understanding, these models exhibit poor language-based spatial reasoning [37]. In fact, they fail at simple tasks such as distinguishing whether an object lies to the left or right of another object (see Table 4.3).

In the case of contrastive language image models (such as CLIP [197], ALIGN [104]), recent works explore how injecting explicit spatial awareness [157, 172, 203, 321] can enable more holistic visual understanding. In fact, [203] shows how such improved spatial awareness benefits model robustness in adversarial domains. This raises the question of how generative language image models, particularly those connecting LLMs to visual encoders [132, 150] can benefit from such spatial

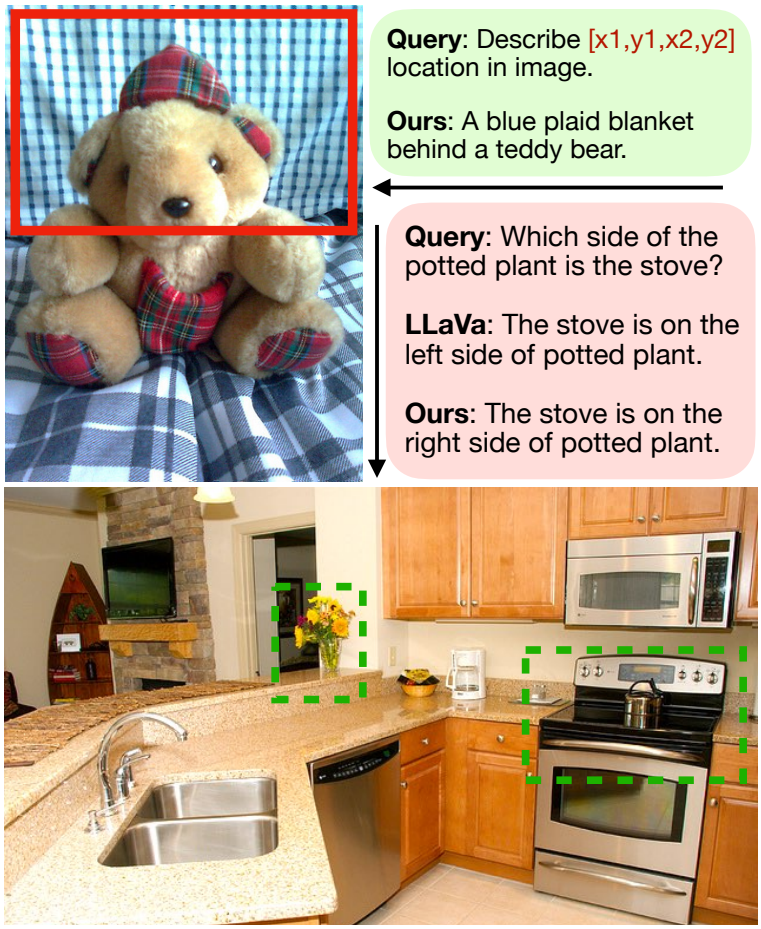

Figure 4.1: We illustrate one unique ability of our model: contextual region description (top). Note the contextual information used in describing the selected region in each image. Explicitly teaching localization to Visual-LLMs also improves spatial awareness in VQA settings (bottom). Color boxes only for illustration purposes.

awareness specific training. We refer to models of this category that generate textual outputs given joint image-text inputs (e.g. [132, 150]) as visual-LLMs (V-LLMs).

In this work, we explore location specific instruction fine-tuning objectives that explicitly enforce V-LLMs to meaningfully process and generate textual image-space coordinates. We hypothesize that such training would lead to improved spatial awareness in these V-LLMs, therein improving performance on VQA tasks. To this end, we propose three instruction fine-tuning objectives that unify location representation with natural language. We also explore optimal representation forms for image-space locations and how pseudo-data generation can be leveraged for efficient scaling of our framework. We name our resulting model as LocVLM.

While the idea of adapting V-LLMs to perform localization related tasks (e.g. detection, segmentation) using V-LLMs has been explored in multiple recent works

[124, 188, 262, 297, 306, 320, 324], these approaches depend on task specific architectural modifications or treat localization inputs / outputs differently from natural language. In contrast, our LocVLM focuses on a unified framework treating location and language as a single modality of inputs with the goal of complementing performance in each task. We intuit that processing location represented in textual form would enforce the LLM to select appropriate image regions as opposed to relying on region level features provided by the architecture. At the same time, textual form location outputs promote spatial awareness at language level in a human interpretable manner, in contrast to using secondary heads or specialized tokens for location prediction. Concurrent work in [37] also explores textual location representation with a generic V-LLM architecture similar to our work. Our proposed LocVLM differs with focus on optimal location representation forms, data-efficient pseudo-labelling, and video domain operation.

Our proposed framework exhibits improved spatial awareness in VQA style conversation demonstrated through experimentation on 14 datasets across 5 vision-language tasks: Spatial Reasoning, Image VQA, Video VQA, Object Hallucination, and Region Description. We summarize our key contributions as follows:

- Inject textual spatial coordinate awareness into V-LLMs
- Propose three novel localization based instruction fine-tuning objectives for V-LLMs
- Discover optimal coordinate representation forms
- Pseudo-Data generation for improved region description and scaling to video domain

4.2 Method

Current V-LLMs [132, 150] exhibit weak understanding of spatial locations within images [37]. We explore and benchmark such shortcomings, and propose three novel instruction fine-tuning objectives aimed at overcoming these drawbacks of existing V-LLMs. We build these objectives based on spatial-coordinate based prompting and demonstrate how LLMs can directly both process and generate meaningful numerical coordinates in image-space after suitable training. In the rest of this section we describe our architecture and training framework, followed by coordinate processing & generation, instruction fine-tuning objectives, pseudo-data generation, and video domain operation.

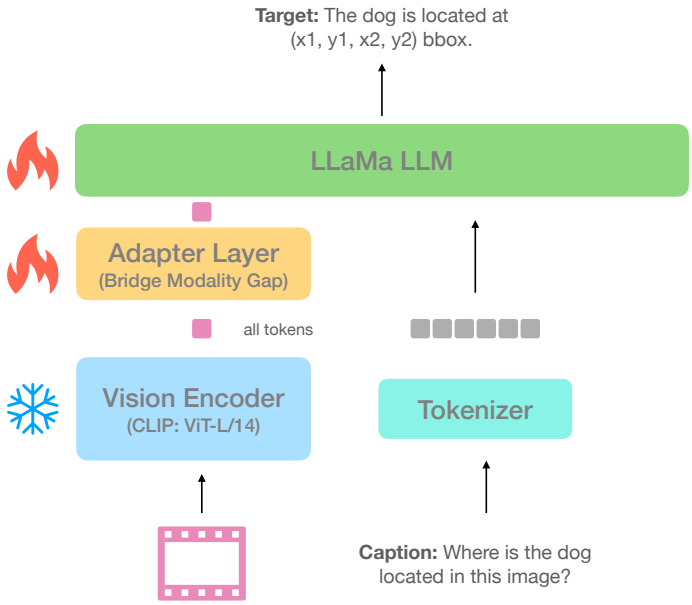

Figure 4.2: **Architecture:** We present the overall model architecture of our framework which is inspired from LLaVa [150].

4.2.1 Architecture and Training

The focus of our work is to explore how spatial localization related training can improve a generic V-LLM such as LLaVA [150]. Therein, our architecture and training framework is inspired from [150]. We use a visual encoder, adapter layer, and LLM stacked sequentially (illustrated in Fig. 4.2), and follow a multi-stage training strategy similar to [150].

Consider an image $X \in \mathbb{R}^{H,W,C}$ where H, W, C $(= 3)$ denote height, width, channels of image and a textual prompt T composed of natural language (asking a question about the image). We define two variants of our model, LocVLM-B and LocVLM-L for better comparison with prior work. We first describe LocVLM-B that processes images with $H = W = 224$. Our visual encoder, ViT-L/14 from CLIP [197], processes the image X to produce a set of 256 visual tokens in $\mathbb{R}^{1024}$, which are in turn projected to $\mathbb{R}^{4096}$ by an adapter layer (implemented as a linear layer). The LLaMA [248] text tokenizer processes the textual prompt T to produce textual tokens in $\mathbb{R}^{4096}$. The joint set of visual and textual tokens (of dimension $\mathbb{R}^{4096}$) are processed by a LLaMA [248] LLM to produce the final set of textual tokens which are in turn untokenized to convert to natural language. The final natural language

CR	GQA (Acc)	RD (METEOR)	A-QA (Acc)
NFP	46.1	19.6	37.1
IVB	47.3	20.7	37.4
DIGA	47.0	20.8	37.3

Table 4.1: Ablation on Coordinate Representation (CR) methods: we compare each of the three proposed CR variants, namely normalized floating point values (NFP), integer valued binning (IVB), and deviation from image-grid based anchors (DIGA).

output is expected to be a suitable response to the input textual prompt, T. In variant LocVLM-L, we use images sized $H = W = 336$ resulting in 576 visual token, an adapter layer implemented as an MLP, and the LLM from LLaMA-2 [249]. All other design choices remain unchanged.

We also highlight the BPE tokenization that is employed in our setup. This learned tokenization scheme may split a single word into sub-parts (that alone can appear meaningless to humans) and handles numerical text (including decimal point) as individual tokens (e.g. 12.34 would be split into 5 separate tokens).

In terms of training, we follow a two-stage strategy. Inspired by LLaVA [150], we adopt an initial pre-training stage that only updates weights of the intermediate adapter layer to align the visual encoder outputs with LLM inputs. Next, we jointly instruction fine-tune the adapter layer and LLaMA LLM with our proposed objectives and template-based localization datasets (see Section 4.3.1). Video domain operation introduces an additional phase (see Section 4.2.5).

4.2.2 Coordinate Processing and Generation

Humans contain the ability to reason about images using image-space coordinates. This is in contrast to existing V-LLMs that can describe the contents of an image elegantly, but lack spatial awareness regarding image contents. We hypothesize that injecting LLMs with additional spatial awareness, through coordinate based reasoning could improve their generic reasoning ability as well. To this end, we introduce our first goal of directly using textual coordinate based image locations in both natural language prompts and LLM generated outputs. For textual coordinates, we explore three different representation forms:

1. Normalized Floating Point Values
2. Integer Valued Binning (across image dimensions)

3. Deviation from Image-Grid based Anchors

For image locations, we explore point based (e.g. center coordinates [cx, cy] of object) and bounding box based (e.g. top-left and bottom-right extreme coordinates of object region [x1, y1, x2, y2]) forms. We next discuss the three representations for coordinates used for either location.

Normalized Floating Point Values calculates absolute image coordinates and normalizes with image dimensions to a (0, 1) range. We use a 4 decimal point representation for these floating point values. While this representation is simple and generic, given the nature of BPE tokenization, each individual coordinate will be represented by up to 6 tokens.

Integer Valued Binning discretizes the absolute image coordinates to one of n_b (=224, 336 for variant B & L respectively) bins spread uniformly across the two image dimensions. Based on the binning parameter, n_b, each coordinate will be represented some number of tokens, in our case up to 3 (less than the floating point variant).

Deviation from Image-Grid based Anchors is motivated from prior object detection works that estimate an initial anchor followed by deviation from that anchor center to estimate bounding box coordinates. We follow a similar setup, where one of n_a anchors is predicted by the model, followed by deviation of coordinate from that anchor center. Our intuition is that, given the sequential next-token prediction setup of LLMs, such a two-stage strategy would lead to faster learning and more accurate coordinates.

We refer to Section 4.5.1 for further details on each variant. In Table 4.1, we ablate each representation format on three different tasks (see Section 4.3.7 for more details) of image VQA (GQA), region description (RD), and video VQA (A-QA). Our experiments indicate optimal performance for integer valued binning (IVB). In all following experimentation, we fix our coordinate representation to IVB.

4.2.3 Instruction Fine-Tuning Objectives

Given suitable coordinate representations, we now have a mechanism to directly prompt LLMs with image locations in textual form. Our second goal is to build training objectives using these image coordinates that directly inject spatial awareness into V-LLMs. We propose three instruction fine-tuning objectives for this purpose.

Let us first revisit the visual instruction fine-tuning methodology in [150]. Building off the COCO dataset, they construct a VQA dataset containing conversa-

Objective	Prompt	Target
LocPred	Where is obj1?	It's at (x1,y1,x2,y2).
NegPred	Where is obj2?	There's no obj2.
RevLoc	Describe (cx,cy)	*Detailed description*

Table 4.2: We summarize our three distinct instruction fine-tuning objectives. Refer to appendix (Section 4.5.2) for exact natural language prompts and targets used for training. For illustration, we use both point and bounding-box based image locations here.

tion style question-answer pairs relevant to each COCO image. Question-answer pairs are generated using an LLM that is fed with the ground-truth bounding-box annotations for each image. Inspired by this setup, we build a similar spatial-VQA dataset using images and annotations of the COCO dataset, but instead of LLM prompting, we utilize hand-crafted templates and pseudo-captions (discussed in Section 4.2.4) to generate conversations.

We propose three types of question-answer pairs that relate to our three instruction fine-tuning objectives: Location Prediction (LocPred), Negative Prediction (NegPred), and Reverse-Location Prediction (RevLoc). See Table 4.2 for examples. Considering the LLM based final text generation in our architecture, we utilize next-token prediction loss to achieve each objective during our training.

Location Prediction: Given an object category, we query the model to generate a point or bounding box localizing that object in the image. The object category and bounding box are derived from the COCO train set annotations. To avoid object mismatches (i.e. multiple object of same category), we first filter images containing only a single object of a given class.

Negative Prediction: Using the same prompt templates as in *LocPred* above, we query the model to generate a point or bounding box localizing a specified object in the image. However, in this case we select an object category not present in the image and accordingly provide a target text of "no such object in image". For each image, we utilize COCO bounding-box annotations to discover objects (belonging to COCO classes) that are not present in that image.

Reverse-Location Prediction: We perform the reverse of *LocPred* here. Given a point or bounding box in image space, we query the model to describe the object in that location. The bounding box and object category are derived from the COCO train set annotations.

While introducing three novel train objectives aimed at injecting location information, we highlight that our proposed framework relies on training data (i.e.

human annotations) identical to those used by LLaVA [150]. We do not use any additional ground-truth annotations for training. Next we explore how we could augment the generality of our framework while limiting to this same annotated data.

4.2.4 Pseudo-Data Generation

We introduced three train objectives, each utilizing template based conversations as prompts and targets. However, our reliance on categories of COCO dataset limits the object vocabulary seen during training. Therein, we propose a pre-trained V-LLM based pseudo-data generation strategy. In fact, we utilize our model after stage one training as the V-LLM leading to a form of self-training based learning. Given the abundance of only image-level annotated datasets (i.e. no bounding box ground-truth), we also explore how an object-detector generated pseudo-bounding boxes could augment our framework.

Self-Training: Given an image and bounding box annotations from the COCO train set, we prompt the V-LLM to caption each distinct object in the image. In order to prevent ambiguous object queries, we filter images to select only those containing at most one instance of a single category. We additionally prompt the V-LLM to describe the object using relational information (i.e. relative to other objects in image). This process provides us a dataset with object level bounding boxes and descriptive captions that are not limited to the COCO object categories (dataset details in Section 4.5.3). In turn, we use this data generated by the V-LLM (our stage one model) to further improve performance of our framework. We modify each of our three train objectives (in Section 4.2.3) to utilize these image-specific pseudo-captions instead of the generic dataset level category labels.

Weak Supervision: We explore how datasets containing no object level annotation (e.g. video classification / VQA datasets) could be leveraged to adapt our framework into domains beyond images. Therein, we utilize an off-the-shelf panoptic segmentation framework from SEEM [331] to generate pseudo-bounding boxes for selected object categories within any image as well as exhaustive pixel level labels (enabling negative class identification). We leverage this setup to extend our introduced train objectives to the video domain as well.

4.2.5 Video Domain Operation

Inspired by the simple modifications to LLaVA [150] in [171] enabling video domain operation, we follow a similar strategy of modifying our LocVLM-B archi-

tecture to process videos while introducing no additional components. The visual backbone process multiple video frames individually (as images) and resulting tokens are averaged across spatial (S) and temporal (T) axes to obtain $S + T$ tokens. These are processed by the adapter layer and LLM to generate the textual outputs. Further details on our video architecture are discussed in Section 4.5.4.

In addition to our two training phases discussed in Section 4.2.1, we introduce a third video instruction fine-tuning stage using a dataset we derive from ActivityNet [87]. Following [171], only the adapter layer is fine-tuned leaving all other parameters frozen. This resulting model is referred to as LocVLM-Vid-B.

We next introduce video variants of our three instruction fine-tuning objectives focused on static objects in videos. We utilize our proposed pseudo-labeling strategy to generate necessary video annotations and train both the adapter layer and LLM to obtain a second video model tagged LocVLM-Vid-B+. Futher details on our video fine-tuning objectives are presented in Section 4.5.4.

4.3 Experiments

In this section, we present experimental results to highlight existing weaknesses of SOTA V-LLMs and how our proposed framework addresses these issues. We also evaluate on standard VQA benchmarks across domains to showcase the better reasoning abilities of our model and highlight novel abilities of our framework.

4.3.1 Experimental Setup

Datasets: We utilize the COCO dataset [144] and our model (post stage one training) to construct a localization related VQA dataset as outlined in Sections 4.2.3 and 4.2.4. We name this dataset *Localize-Instruct-200K*. In detail, this contains LocPred and RevLoc question-answer pairs that use pseudo-captions instead of COCO categories as well as NegPred. We define a second video dataset, *Localize-ActivityNet*, containing question-answer pairs constructed from ActivityNet pseudo-bounding boxes following Section 4.2.5. Our models are primarily trained on our Localize-Instruct-200K dataset. Our stage one training uses CC3M dataset [35]. Additionally, our Localize-ActivityNet dataset and ActivityNet dataset [87] are used for video domain training.

Training: We train our models on 8xA100 GPUs (each 80GB) following a two-phase training schedule. Our first phase trains on CC3M [35] following the setup in [150]. The second phase uses our Localize-Instruct-200K dataset and trains for 10 epochs with a batch size of 64, ADAM-W optimizer with initial learning

rate $2e-5$, 0.3 warm-up ratio, and cosine-decay learning rate schedule. Both the training phases we conduct use standard next-token-prediction loss used in LLM training.
Evaluation: During evaluation, following standard protocol [150], we iteratively generate next tokens, given visual and textual inputs. The LLM output is a distribution across the entire token vocabulary. The next token is selected through multinomial sampling of this output using a softmax temperature term of 0.2 during normalization.

4.3.2 Spatial Reasoning: A Toy Experiment

We investigate spatial reasoning abilities of two SOTA V-LLMs, LLaVA [150] and BLIP-2 [132], using a simple toy experiment. We create an evaluation dataset from COCO annotations containing images with distinct category object triplets (only one instance occurrence of each object category), where each object is entirely to the left or right half of the image and two objects are on opposite sides. The ground-truth bounding box annotation are utilized to automate this dataset creation procedure. This evaluation set, referred as *COCO-Spatial-27K*, contains 26,716 image-question pairs (see Section 4.5.3 for details). We introduce two evaluation settings, direct VQA and in-context learning (ICL) VQA to understand spatial reasoning abilities of these models. In direct VQA, given an image we query the model whether an object lies above or below another object. In ICL VQA, before a similar final query, we provide two example question-answer pairs (involving the other two objects in the image) in the same format as our query. Refer to Section 4.5.5 for further details on task. We perform the same for objects in top vs bottom halves of images.

These results are presented in Table 4.3. Our results indicate near random performance for existing V-LLMs. For the case of LLaVA, we perform keyword (*left* and *right*) frequency analysis on its instruction tuning dataset (LLaVA-Instruct-80K dataset) to verify the presence of terms *left* and *right* in its training corpus. These keywords are present in 0.37% and 1.13% of its conversations respectively (see Section 4.5.6 for more) indicating presence of these concepts in the image-text training corpus. In contrast to these methods, our proposed framework notably improves performance over both BLIP-2 [132] and the LLaVA baseline [150].

Method	ICL	All	Left	Right	All	Above	Below
BLIP-2 [132]	✗	45.5	86.1	4.74	49.2	50.4	48.6
LLava [150]	✗	55.1	84.5	36.5	58.9	57.8	59.3
Ours	✗	69.5	79.7	59.2	65.4	64.2	65.9
BLIP-2 [132]	✓	14.7	17.8	11.6	15.8	16.5	15.2
LLaVa [150]	✓	55.1	84.7	36.4	58.2	57.7	58.5
Ours	✓	76.5	90.4	61.5	74.1	73.5	74.4

Table 4.3: **Spatial Reasoning**: We report accuracy (%) on a spatial localization dataset derived from COCO annotations to highlight weak spatial awareness of existing V-LLMs. We query these models to answer whether one object is to the left or right / above or below of another object. The SOTA V-LLMs evaluated exhibit close to random performance. Our proposed setup outperforms existing methods. Ours refers to LocVLM-B variant.

4.3.3 Image VQA

Image VQA involves correctly answering natural language questions regarding content within an image. We evaluate our model for Image VQA on two standard datasets, GQA and VQAv2. The GQA dataset focuses on questions requiring compositional reasoning, particularly involving surrounding information of objects within an image. We evaluate on its test-dev split containing 12,578 image-question pairs. The VQAv2 dataset contains open-ended questions about each image that require an understanding of vision, language and commonsense knowledge to answer. We use its validation split containing 214,354 image-question pairs for our evaluation. For each dataset, we follow standard V-LLM evaluation protocol following [132, 171] and report top-1 accuracy metric. Our results in Table 4.4 indicate clear improvements for LocVLM over our baseline and prior work, establishing the usefulness of our proposed framework. The closest to our work, Shikra [37] achieves performance competitive to our LocVLM-B, but unlike ours they use VQA datasets (containing similar domain question-answer pairs) during training.

4.3.4 Video VQA

Our model is also applicable to video tasks following our video domain adaptation described in Section 4.2.5. We simply adopt the additional video instruction fine-tuning phase from [171] on the ActivityNet dataset after our initial two phases of training to obtain LocVLM-Vid-B. This third phase involves fine-tuning only the

Method	LLM	VS	Zero-Shot	GQA	VQA-V	VQA-T
SR [15]	-	-	✗	62.1	72.9	-
Shikra [37]	7B	224	✗	-	75.3	77.4
LLaVA-v1.5	7B	336	✗	62.0	78.1	78.4
LocVLM-L	7B	336	✗	**63.5**	**78.2**	**78.6**
LLaVA-v1	7B	224	✓	44.7	49.8	49.3
LocVLM-B	7B	224	✓	**47.3**	**50.3**	**50.8**
Viper-GPT	175B	-	✓	48.1	-	-
BLIP-2	11B	-	✓	44.7	54.3	53.9
LLaVA-v1.5	7B	336	✓	48.7	55.7	55.3
LocVLM-L	7B	336	✓	**50.2**	**55.9**	**56.2**

Table 4.4: **Image VQA Results**: We report accuracy (%) on the test-dev split of GQA dataset (GQA) and the validation / test splits of VQAv2 dataset (VQA-V / VQA-T). Our proposed LocVLM improves over prior works achieving state-of-the-art performance.

adapter layer of our model. We also explore video variants of our IFT objectives that train both adapter layer and LLM. The resulting model is termed LocVLM-Vid-B+.

Video VQA focuses on correctly answering questions regarding a given video that require spatio-temporal awareness to answer. We evaluate our video-adapted model on the task of zero-shot video VQA on four benchmark datasets, ActivityNet-QA, MSRVTT-QA, MSVD-QA, and TGIF-QA. We evaluate on the validation splits of these four datasets. ActivityNet-QA videos cover a wide range of complex human activities relevant to daily living with its question-answer pairs focusing on long-term spatio-temporal reasoning. MSRVTT-QA builds off the MSRVTT dataset that contains web videos covering a comprehensive range of categories and diverse visual content. MSVD-QA is a similar dataset building off the MSVD dataset. TGIF-QA contains question-answer pairs from an dataset constructed of animated GIFs. For each dataset, we report the accuracy metric following evaluation protocol in [171]. Our results on these four datasets reported in Table 4.5 demonstrate state-of-the-art performance of our proposed LocVLM-Vid-B, with consistent improvements over the baseline from [171]. Here we use the LocVLM-Vid-B variant for fairer comparison with the baseline from [171]. We attribute the performance gains exhibited by our model to its stronger spatial awareness (see Section 4.3.2). Particularly in the case of video understanding, awareness of content at spatial level of each frame is significant to understand object motions and interactions [2, 218]. We also report more results involving additional model variants in Table 4.6.

Method	Zero-Shot	ActivityNet-QA	MSRVTT-QA	MSVD-QA	TGIF-QA
JustAsk [291]	✗	38.9	41.8	47.5	-
FrozenBiLM [293]	✗	43.2	47.0	54.8	-
VideoCoCa [289]	✗	56.1	46.3	56.9	-
Flamingo [5]	✓	-	17.4	35.6	-
BLIP-2 [132]	✓	-	17.4	34.4	-
InstructBLIP [50]	✓	-	25.6	44.3	-
FrozenBiLM [293]	✓	24.7	16.8	32.2	41.0
Video Chat [134]	✓	26.5	45.0	56.3	34.4
LLaMA Adapter [319]	✓	34.2	43.8	54.9	-
Video LLaMA [316]	✓	12.4	29.6	51.6	-
Video-ChatGPT [171]	✓	35.2	49.3	64.9	51.4
LocVLM-Vid-B	✓	**37.4**	**51.2**	**66.1**	**51.8**

Table 4.5: **Video VQA Results**: Our proposed LocVLM-Vid-B improves over Video-ChatGPT [171] and achieves state-of-the-art results (Top-1 Accuracy %) across four different video VQA benchmarks. Note the zero-shot setting of all these evaluations.

4.3.5 Object Hallucination

Current state-of-the-art V-LLMs suffer from object hallucination, generating image descriptions inconsistent with the image content [140]. For example, a V-LLM would respond to “Where is the cat in this image?" with “The cat is on the table" when in reality there is no cat in the image. We evaluate the extent of hallucination in V-LLMs using three datasets we introduce (details in Section 4.5.3) and the POPE dataset [140]. Our three datasets, Hal-COCO, Hal-ADE, and Hal-Act build off COCO, ADE-20K, and ActivityNet datasets respectively. The first two involve images and the latter videeos. These datasets contain ‘Is there *obj* in image / video?” type questions per sample, for two objects present and not present in the image / video. Hal-ADE object categories contain *no overlap* with COCO classes allowing evaluation on novel object categories unseen during our instruction fine-tuning. Results reported in Table 4.7 show clear improvements of LocVLM-B over baselines. We also evaluate LocVLM-B on the POPE benchmark [140] that builds off the COCO dataset object annotations and report results in Table 4.8. Our LocVLM showcases similar performance improvements on this dataset.

Method	VLT	Frames	Acc (%)
LLaVa (v1) [150]	✗	1	28.7
LLaVa (v1.5) [150]	✗	1	31.5
LocVLM-B	✗	1	29.2
LocVLM-L	✗	1	**32.1**
Video-ChatGPT [171]	✓	100	35.2
LocVLM-Vid-B	✓	100	37.4
LocVLM-Vid-B+	✓	8	**38.2**

Table 4.6: **Video VQA**: We report more results (Top-1 Accuracy) for ActivityNet-QA dataset including multiple baseline and LocVLM variants. Our proposed models exhibit top performance. VLT denotes video level training. More details in Section 4.5.4.

Method	Hal-COCO	Hal-ADE	Hal-Act
Shikra [37]	86.2	58.7	-
LLaVa [150]	61.9	53.8	-
LocVLM-B	**88.3**	**75.2**	-
Video-ChatGPT [171]	-	-	50.6
LocVLM-Vid-B	-	-	68.7
LocVLM-Vid-B+	-	-	**72.4**

Table 4.7: **Hallucination Evaluation**: We report top-1 accuracy (%) for object presence type questions and showcase reduced object hallucination in our proposed framework.

4.3.6 Region Description

A unique characteristic of our model (in contrast to V-LLMs like LLaVA [150] & BLIP-2 [132]) is its ability to reason with prompts involving coordinate based image space locations without any input modifications. Given a point or bounding box location, we prompt our model to generate an output describing that location. We refer to this unique ability of our model as *region description* (RD). We evaluate this RD capability of our model by generating object level descriptions focused on contextual information (e.g. surrounding of that object in the image). Following evaluation protocol in [188] for region description, we extend their evaluation to three standard referring localization datasets from RefCOCO [113] and report these results in Table 4.9. We select the METEOR score as the evaluation metric

Datasets	Metrics	BLIP-2	Shikra	LLaVA	Ours
Random	Accuracy (↑)	88.6	86.9	50.4	87.9
	Precision (↑)	84.1	94.4	50.2	83.6
	Recall (↑)	95.1	79.3	99.1	93.9
	F1 Score (↑)	**89.3**	86.2	66.6	88.5
	Yes	56.6	43.3	98.8	56.2
Popular	Accuracy (↑)	82.8	84.0	49.9	86.0
	Precision (↑)	76.3	87.6	49.9	79.7
	Recall (↑)	95.1	79.2	99.3	93.9
	F1 Score (↑)	84.7	83.2	66.4	**86.3**
	Yes	62.4	45.2	99.4	58.9
Adversarial	Accuracy (↑)	72.1	83.1	49.7	78.8
	Precision (↑)	65.1	85.6	49.9	76.6
	Recall (↑)	95.1	79.6	99.1	93.7
	F1 Score (↑)	77.3	82.5	66.3	**84.3**
	Yes	73.0	46.5	99.4	61.7

Table 4.8: **More object hallucination:** Results on POPE evaluation benchmark [140] indicate strong performance of our model.

to account for variations in word choice in generated answers which may be acceptable in various cases (e.g. different sentence structure leading to alternate word ordering). Our results indicate clear improvements over the LLaVA baseline [150] as well as prior state-of-the-art. We attribute these improvements to our pseudo-data based training.

4.3.7 Ablations

Next we conduct ablative studies on separate components of our proposed setup: IFT objectives, location type, and pseudo-data. We follow the same training strategy as described in Section 4.3.1 and present these results in Table 4.10. LocVLM-B is used for all these experiments. The significance of each IFT objective is verified in Table 4.10 (top) with consistent performance improvements across tasks. The generality of our approach to differing location type (i.e. points vs bounding boxes) and usefulness of pseudo-data is visible in Table 4.10 (bottom). In particular, we highlight the notable performance improvement for RD task gained from using pseudo-data. We also conduct ablations for our video domain training setup and report these results in Table 4.11. The LocVLM-Vid-B+ variant is used in these experiments. Our results showcase the usefulness of proposed IFT objectives

Method	ZS	RefCOCO	RefCOCO+	RefCOCOg	
				Val	Test
SLR [299]	✗	-	-	-	15.4
SLR + Rerank [299]	✗	-	-	-	15.9
Kosmos-2 [188]	✗	8.67	8.82	14.3	14.1
Shikra [37]	✗	10.4	11.1	19.7	19.5
LLaVa [150]	✗	8.43	8.73	13.5	13.5
LocVLM-B	✗	**14.6**	**15.2**	**26.0**	**26.2**
Kosmos-2 [188]	✓	6.34	8.25	12.4	12.2
LLava [150]	✓	4.23	7.26	10.6	10.3
LocVLM-B	✓	**11.0**	**11.1**	**20.6**	**20.7**

Table 4.9: **Region Description:** We report METEOR scores for RD task [188]. Test-B split is used for RefCOCO & RefCOCO+ datasets. Our method outperforms all prior work.

for video domain learning as well.

4.4 Conclusion

We introduce a simple framework that equips visual-LLMs (V-LLMs) with greater spatial understanding, termed LocVLM. We leverage the idea of encoding image coordinates within language to propose three instruction fine-tuning (IFT) objectives. This training process endows V-LLMs with the ability to reason about spatial composition of images using image space coordinates within text. A data efficient training pipeline utilizing pseudo-data allows our approach to achieve state-of-the-art results in Image VQA, Video VQA, and Region Description while improving spatial awareness and reducing object hallucination.

4.5 Additional Details

4.5.1 Coordinate Representation Details

We describe our three coordinate representation variants in detail, first focused on bounding-box location format . Consider an image of dimensions (512, 512) containing a cat. Let $(10, 120, 30, 145)$ define the minimal bounding box enclosing the cat in image space ordered as (x1,y1,x2,y2) where (x1,y1) would describe the

LocPred	NegPred	RevLoc	GQA	RD	A-QA
✗	✗	✗	44.7	10.3	35.2
✓	✗	✗	45.2	12.2	35.8
✓	✓	✗	46.9	12.5	37.2
✓	✓	✓	47.3	20.7	37.4

Location Type	PD	GQA	RD	A-QA
Point	✓	47.3	20.6	37.4
Bounding Box	✓	47.3	20.7	37.4
Bounding Box	✗	46.5	11.6	37.1

Table 4.10: **Ablations:** We report top-1 accuracy (%) on GQA and ActivityNet-QA (A-QA) datasets and METEOR scores for RD task on RefCOCOg test split. (top) We ablate proposed instruction fine-tuning objectives to verify usefulness of each objective. (bottom) We first ablate point based and bounding box based location forms to showcase minimal difference across them. We next ablate use of object description pseudo-data (PD). We highlight the improvements due to pseudo-data, especially on the RD task.

top left corner and (x2,y2) would describe the bottom right corner of that bounding box. We will use this example in following explanations.

Normalized Floating Point Values would normalize these coordinates using image dimensions to a (0,1) range and directly use normalized values rounded to 4 decimal places. In the given example, the location of the cat would be described $(0.0195, 0.2344, 0.0586, 0.2832)$ which is equal to $(10/512, 120/512, 30/512, 145/512)$ after appropriate rounding.

Integer Valued Binning considers n_b fixed bins across the image that are described by integers 0 to n_b. In our case, for the LocVLM-B version we fix n_b to 224 and for LocVLM-L version we fix n_b to 336. The original bounding-box coordinates are mapped to the range $(0, n_b)$ inspired by prior work [41, 262] using similar binning strategies. In the case of our examples, the location of the cat would be described $(4, 52, 13, 63)$ for $n_b = 224$ which can be easily calculated by remapping the coordinate range as $(n_b \cdot 10/512, n_b \cdot 120/512, n_b \cdot 30/512, n_b \cdot 145/512)$ with integer rounding.

Deviation from Image-Grid based Anchors defines a grid of anchors in image space, selects the anchor closest to the object center, and measures each bounding box coordinate as a deviation from that anchor center. In our case, we set $n_a = 16^2$ for LocVLM-B and and $n_a = 24^2$ for LocVLM-L (motivated by the visual encoder

LocPred	NegPred	RevLoc	A-QA
✗	✗	✗	37.4
✓	✗	✗	37.6
✓	✓	✗	38.2
✓	✓	✓	38.2

Table 4.11: **Video Ablation:** We report top-1 accuracy (%) on ActivityNet-QA (A-QA) dataset. Results indicate the generality of proposed IFT objectives for video domain training as well.

transformer grid size). In both cases, each anchor covers a 14×14 pixel patch. We describe the anchors using (p, q) for $p, q = 0, 1, ...13$. For our example, the bounding box fits the anchor $(0, 4)$ and we represent the bounding box as $(0, 4, 3, 11, 6, 0)$ where the latter four values correspond to pixel deviations from the selected anchor center located at $(7, 63)$ in (224×224) image space.

We also utilize the alternate location form of point values, i.e. (cx, cy) for object center coordinates in image space. Coordinate representations are utilized in the same manner. Instead of four coordinates, we only use two that correspond to the object center. For our given example, the center of the cat would be $(20, 132.5)$ which would be represented similar to the bounding box case.

4.5.2 Training Prompt Details

We introduce three instruction fine-tuning objectives that utilize specific hand-crafted templates to generate the target prompts used during training. We discuss in detail, these three objectives presented in Table 4.2 (main paper): LocPred, NegPred, and RevLoc.

For the first two cases, we use a set of 5 templates, one of which is randomly selected for each sample during training.

1. `Where is the object described {category} located in image in terms of {repr}?`
2. `What is the location of object described {category} in terms of {repr}?`
3. `Localize the object described {category} in terms of {repr}?`
4. `Provide a {repr} for the the object described {category}?`
5. `Generate a {repr} for the the object described {category}?`

The placeholder `{category}` is replaced with the relevant ground-truth annotation of each particular object. In the case of COCO dataset, these correspond to one

of the 80 COCO categories. For Localize-Instruct-200K (our constructed pseudo-caption dataset), the object pseudo-description is used in place of {category}. The {repr} can be one of rep_bbox = (x1,y1,x2,y2) bbox or rep_point = (cx,cy) point.

For LocPred, the target is of form "It is located at {loc}" while for Neg-Pred, the target is "There is no such object in the image". The same five identical prompts are randomly assigned to each objective to ensure no input patterns allow distinguishing between the two targets.

For the case of RevLoc, we similarly sample one prompt from the following set of 3 templates:

1. Describe the object located at {loc}?
2. Provide a caption for object at {loc}?
3. What is at location {loc} in image?

The target is of form "There is a {category}." where category can either be class label or a pseudo-description of that location.

4.5.3 Dataset Details

In our work, we first perform blurring of human faces across all our data to preserve privacy in resulting models. These modifications are applied to all our datasets before performing any model training.

As described in Section 4.2.4 (main paper), we explore pseudo-data generation to construct two new datasets, one for object level captions in images and the other for video object labels. We name them first PRefCOCO-100K, and utilize it to construct our Localize-Instruct-200K dataset used for our image level instruction fine-tuning (IFT) objectives. We name the second Pseudo-ActNet and utilize it in our video level IFT objectives.

PRefCOCO-100K uses 95899 images from the COCO dataset and uses an image VQA model (LLaVa [150]) to generate object level descriptions using the COCO object annotations. We first filter images to select those containing unique instances of objects (e.g. only one dog in the image as opposed to multiple dogs). This results in the 95899 images. Next, we ask the VQA model to generate a suitable caption that describes the object category using both its characteristics and relations to surrounding. In detail, we use the exact prompt "Describe the {category} in this image using one short sentence, referring to its visual features and spatial position relative to other objects in image." where category is the ground-truth object label. These obtained object-level captions are used to create question-answer (QA) pairs for the images, resulting in 402,686 such QA pairs.

Following the prompting mechanisms for LocPred and RevLoc described in Section 4.5.2, we generate image-conversation pairs from PRefCOCO-100K, resulting in a human-conversation style dataset we use for training. We refer to this dataset as Localize-Instruct-200K. This contains twice as many image-conversation pairs as the original, given repeated images for both LocPred and RevLoc objectives. This is the main dataset used for our image level training.

For our video domain IFT objective based training, we only use category level labels and leave caption level training as a future direction. We construct Pseudo-ActNet dataset that contains generated bounding-box annotations for all objects belonging to COCO panoptic segmentation dataset [144] categories. Eight uniformly sampled frames are processes per video for annotation. We utilize the pre-trained SEEM [331] model (motivated by [140]) to generate pixel-level panoptic segmentation outputs for each selected frame and convert these segmentations to bounding boxes (panoptic also contains instance level distinction allowing straightforward bounding box extraction). The panoptic outputs (label for each pixel) also allows to obtain an exhaustive list of all COCO dataset categories present in each video - this is necessary to find suitable negative categories for our NegPred objective. Therein, for 8 uniformly sampled frames of each video in the ActivityNet train split, we generate bounding box annotations for all objects belonging to COCO dataset categories and a list of COCO dataset categories not present in those 8 frames. This data is sufficient to implement our IFT objectives on the ActivityNet video dataset with only the videos from the dataset. Our promising results (see Table 4.6) for video-domain IFT using only pseudo-data highlight the data scalability of our proposed framework.

4.5.4 Video Architecture & Training

As discussed in Section 4.2.5 (main paper), we introduce two video-domain variants of our framework, LocVLM-Vid-B and LocVLM-Vid-B+. We first detail the architecture common to both variants, followed by specific training procedures.

The overall architecture remains consistent to what is presented in Fig. 4.2. The visual encoder processes n_f frames independently as images to produce $n_f \times 256$ visual tokens per video (where 256 is tokens generated per image). The spatio-temporal pooling strategy from [171] is utilized to obtain a set of $256 + n_f$ visual tokens per video. In detail, the visual tokens are average pooled across the temporal dimension to obtain 256 spatial tokens and across the spatial dimensions to obtain n_f temporal tokens. These are concatenated to obtain the $256+n_f$ visual tokens per video. The adaptor layer and LLM remain unchanged - this is straightforward since

both these layers perform set-to-set operations independent of input sequence length.

The LocVLM-B-Vid+ variant combines our video level IFT objectives with the training setup from [171]. Given early experiments suggesting insufficiency of fine-tuning only the adapter layer for our IFT objectives, we fine-tune both the LLM and the adaptor layer. We also sample only 8 uniformly spaced frames per video (for compute reasons). The three IFT objectives are modified to suit video domain operation. Given the lack of explicit temporal modelling in our visual backbone and the limited spatio-temporal awareness even within the LLM, we focus on static objects in videos to construct IFT targets. For LocPred and RevLoc, we first filter out objects to select those present only in one of the eight frames or relatively static ones (bounding-box center (x,y) is within a 5 pixel range from their average if present in multiple frames). Then, we obtain the average bounding-box for that object across the frames. These static bounding boxes and negative categories (from the dataset) are used to construct the IFT targets in the same manner as we do for images.

4.5.5 Spatial Reasoning Toy Experiment

We present additional details of the toy experiment introduced in Section 4.3.2. We describe the dataset used for evaluation, templates for prompting, and evaluation metric calculation. We also repeat our results from Table 4.3 (main paper) for the left vs right variant here in Table 4.12.

We first construct an evaluation dataset, tagged *COCO-Spatial-27K* containing 26,716 image-question pairs. We build this off the COCO dataset [144] train split through a fully-automated process, utilizing the ground-truth object bounding-box annotations. We first filter out images based on three constraints - this eliminates a large portion of images; hence we elect to use the train split to obtain a considerable quantity of samples after filtering. We first select images containing distinct category object triplets (only one instance occurrence of each object category). For example, an image would contain categories person, dog, and table but only one of each. The second constraint ensures that each object is entirely to the left or right half of the image. This is based on object center not being in the central 20% region. The third constraint is that at least two objects are on opposite sides (i.e. left and right half of image). This provides at least two opposite side object pairs. The ground-truth bounding box annotations enable easy automation of this filtering procedure.

We next discuss our templates for prompting. For two objects on opposite sides

Method	ICL	Acc (All)	Acc (Left)	Acc (Right)
BLIP-2 [132]	✗	45.5	86.1	4.74
LLaVA [150]	✗	55.1	84.5	36.5
Ours	✗	69.5	79.7	59.2
BLIP-2 [132]	✓	14.7	17.8	11.6
LLaVA [150]	✓	55.1	84.7	36.4
Ours	✓	76.5	90.4	61.5

Table 4.12: **Spatial Reasoning**: We repeat our results for left vs right objects here.

tagged `obj_1` and `obj_2`, we use the prompt `Which side of obj_1 is obj_2 located?` and query the model for a response. This is for the direct VQA setting. In the case of in-context learning (ICL) VQA setting, we preprend two examples to the prompt: `Q: Which side of obj_1 is obj_2 located? A: The obj_1 is located to the left of obj_2. Q: Which side of obj_2 is obj_1 located? A: The obj_2 is located to the right of obj_1. Q: Which side of obj_3 is obj_1 located?`. In this case, `obj_3` is the third object, and their ordering is selected such that `obj_1` is on one side, and `obj_2`, `obj_3` are on the opposite side.

Building off standard VQA protocol in [97, 171], we simply query is the terms `left` or `right` are present in the generated outputs, and rate it a success is the target term is present in the generated response. We also visualize some examples for this task in Fig. 4.3.

4.5.6 LLaVA Dataset Analysis

Our results in Table 4.12 indicate unusual disparity in left vs right accuracy numbers, especially in LLaVA [150]. We analyse the training dataset used in this LLaVA baseline to better understand these disparities.

The LLaVA model [150] is instruction fine-tuned on a human conversation style dataset (LLaVA-Instruct-80K). This dataset contains 80,000 image-conversation pairs leading to 221,333 question-answer (QA) pairs across all images (multiple QA for single image). We analyse the presence of keywords related to `left` and `right` concepts that are probed in our spatial-reasoning toy experiment (Section 4.3.2).

We first analyse the exact presence of the words *left* and *right* in the corpus (noting this maybe in different context, e.g. who has the right of way?). Of the 80,000 image-conversation pairs, left and right are present in 1619 (2.02%)

and 5001 (6.25%) cases respectively. We provide further statistics of the dataset in Table 4.13 indicating some presence of conversation style training samples encompassing left & right concepts. A large count of the keyword `right` occurs in contexts with different meanings while `left` mostly occurs in its spatial context. We hypothesize that this may be the reason for predicting `left` more often when models are queried with a spatial reasoning related question (i.e. keyword `left` occurs more frequently with *spatial related words* in training corpus).

Template	Left (%)	Right (%)
"the {} "	171 (0.21)	1314 (1.54)
"{} side"	75 (0.093)	110 (0.14)
"to the {}"	80 (0.10)	93 (0.12)

Table 4.13: We count occurrences of various textual phrases related to left & right concepts in the LLaVA-Instruct-80K dataset.

Therein, we attribute these observed disparities for left vs right accuracy numbers to these artifacts present in datasets used for training underlying LLMs.

4.5.7 Limitations & Broader Impact

Our video variant achieves strong performance on VQA tasks but fails to understand temporal locations. In fact, direction use of temporal locations paired with spatial locations results in training collapse for our framework. Extension of our instruction fine-tuning objectives to suitably utilize time coordinates is left as a future direction. In terms of broader impact, while our model uses generic vision and language model architectures, we note that our training data from public datasets may contain biases which should be taken into account when deploying models trained using our framework.

4.5.8 Qualitative Evaluation

In this section, we present visual examples showcasing various aspects of our frameworks capabilities. We broadly consider the three distinct settings of spatial reasoning, region description, and generated locations. Note that in all visualizations we blur human faces to make them unidentifiable for privacy reasoings.
Spatial Reasoning: We illustrate examples from our COCO-Spatial-27K dataset highlighting both success cases and failures of our framework. These qualitative results are presented in Fig. 4.3. In each case, let as tag the two objects

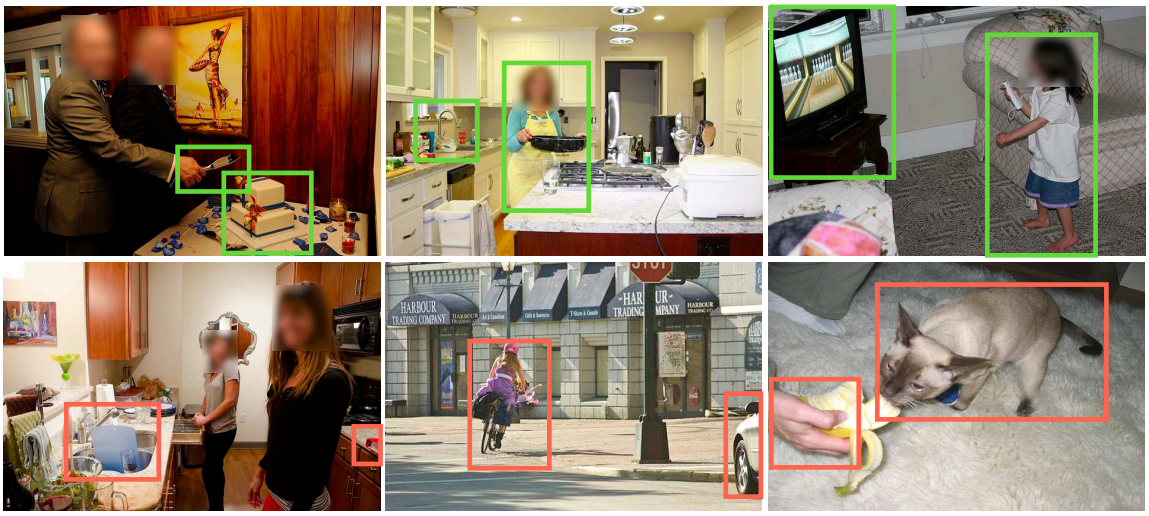

Figure 4.3: Visualizing Spatial Reasoning: We illustrate example images on which we perform our toy experiment for spatial reasoning (Section 4.5.5). Success cases on top row (green) and failure cases on bottom row (red).

within bounding boxes as `obj1` and `obj2`. Following Section 4.5.5, we prompt our framework with each image and `Which side of obj1 is obj2?` and match the response with the ground-truth answer. Correct matches (success cases) on presented on the top row (green) and incorrect matches (failure cases) on bottom row (red). The correct matches indicate the spatial reasoning abilities of our framework across a wide range of image types, including cluttered scenes. The failure cases possibly indicate difficulty at handling truncated / occluded objects.

Region Description: We next illustrate the region description abilities of our model (see Section 4.3.6 for details) in Fig. 4.4. We query our framework with a set of bounding box coordinate such as `Describe the object located at [22, 114, 86, 154]?` (prompt details in Section 4.5.2) paired with each image. We illustrate the object coordinates as a bounding box (green) in each image. The response of the model presented underneath each image. We highlight invalid responses in red. These qualitative evaluations indicate the ability of our model to not only detect the object present in the queried region, but also describe it in terms of its surrounding: an ability unique to our model in contrast to traditional object classifiers or detectors. At the same time, the generated responses display limitations in terms of object characteristic hallucination and minimal spatial relation (e.g. to the left / right of) based description.

Generated Locations: In our experiments, the tasks of object hallucination and

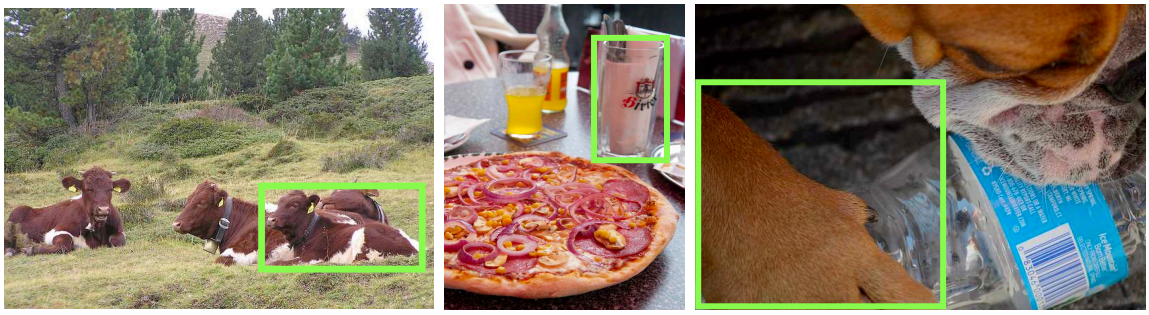

Figure 4.4: Visualizing Region Description: Our framework possesses the unique ability of generating representative descriptions for a selected region of an image, input to the model in terms of textual coordinates. We illustrate 3 example images with a bounding box (green) denoting the queried region. The responses generated by our model are underneath each image, with invalid outputs highlighted red.

region description directly evaluate the learning resulting from IFT objectives NegPred and RevLoc respectively. In this section, we present some qualitative evaluation to understand the learning resulting from the LocPred objective. These results are visualized in Fig. 4.5. First, these images present samples from the validation split of COCO modified in a similar manner (i.e. filtering explain in Section 4.3) to our training set for LocPred objective. Each image contains one instance of a particular category. The category is labelled on top of each image, and the ground-truth annotation for the object is in green while the prediction by our framework is in blue. We illustrate the success cases of our model in the top row and failure cases in the bottom row. The success cases indicate strong localization skills across diverse scene involving objects of variable sizes. The failure cases denote difficulty in handling crowded / cluttered scenes and truncated / occluded objects. We also note that direct comparison to classical object detectors is unfair given the down-sampled images (i.e. 224×224 or 336 sized) used by our framework (object detectors use higher resolution images).

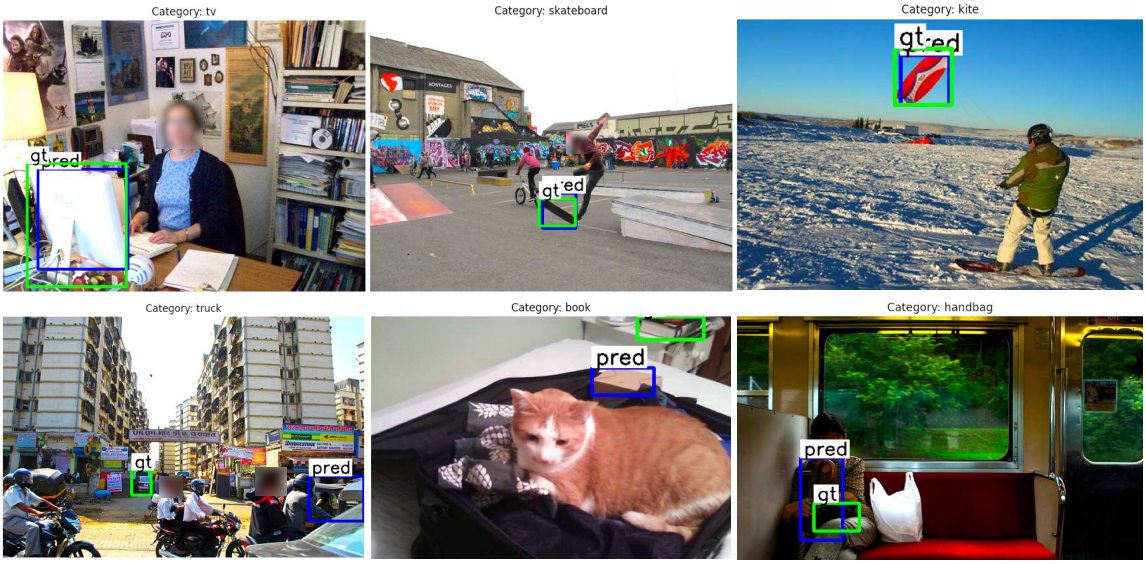

Figure 4.5: Visualization of LocPred Objective: We illustrate the bounding box locations generated by our framework (blue) when queried with a category label (top of each image) and compare with the ground-truth bounding boxes (green). Success cases on top and failure cases on bottom.

Chapter 5

Understanding Long Videos with Multimodal Language Models

5.1 Introduction

> What can we learn from videos,
> beyond scene context understood from a single natural image?

Recent success of large language models (LLMs) and their visual extensions, vision-language models (VLMs), has led to incredible performance on complex language-tied video understanding benchmarks [314], particularly on long-video question answering: a task that requires awareness over longer temporal windows [162] as well as causal and temporal action reasoning [276]. However, the LLMs underlying these approaches contain extensive world knowledge (e.g. understanding of physics, culture, human common sense) and reasoning abilities [265, 298], raising the question of whether they excel at video tasks due to actual *video modality* awareness or simply utilizing world knowledge and contextual information. Such understanding of model reasoning is important for robust deployments avoiding spurious correlation based predictions as well as for better model interpretability [280, 305].

In this work, we systematically study this question in the context of video question-answering (QnA) benchmarks, building two modality-constrained baselines to highlight our findings. These two frameworks are tagged *Just-LLM* and *Single-Frame-VLM*. The first is constrained to access only the task textual query (i.e. no task-specific visual information). The latter is given access to task context

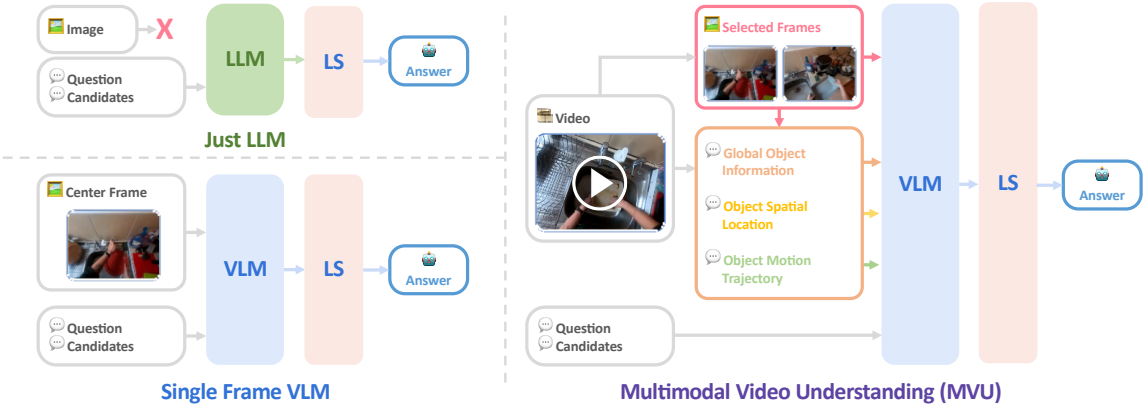

Figure 5.1: **Overview of Framework:** We propose three variants of our framework that solves complex long-video question-answering tasks. (left-top) Just-LLM utilizes only world knowledge with zero task-specific awareness. (left-bottom) Single-Frame-VLM processes an additional center frame to obtain task context but accesses no *video* specific information. (right) Our complete approach, MVU extracts three additional object-centric information modalities followed by fusion in language space. LS refers to likelihood selection.

with an additional single center-frame from the video as input. We discover how these models perform significantly better than random prediction on multiple long-video understanding benchmarks (see Table 5.1, similar findings in [167]). In fact, the latter, utilizing purely world knowledge and contextual information, even outperforms multiple recent state-of-the-art video understanding works (see Table 5.2), challenging the notion of how much *video information* is actually utilized by existing approaches to solve these complex video QnA tasks.

We next focus on efficient inference to allow rapid experimentation with our LLM based frameworks. Therein, we explore suitable prompting and templating to adapt likelihood selection techniques from prior work [215] to video QnA tasks. Our resulting framework achieves more efficient inference with improved performance in comparison to prior work that commonly use auto-regressive generation to tackle long-video QnA benchmarks [13, 264, 314].

Motivated by our initial findings on modality-constrained performance, we study how to inject additional video-specific information into our framework using natural language in a concise and interpretable manner to further improve video understanding. We explore three forms of *object-centric* information modalities, develop pipelines requiring zero video-level training to extract such information using off-the-shelf vision tools, and utilize natural language to fuse this multi-

modal information using templating operations. Our resulting approach, termed Multi-Modal Video Understanding (MVU) framework, while achieving state-of-the-art zero-shot performance across long-video understanding benchmarks, also exhibits better interpretability (e.g. exposing video-specific information utilized) through its language-based operation. Moreover, our proposed MVU exhibits generality with its strong performance even on robotics domain tasks.
In summary, our key contributions are as follows:

1. Uncover surprisingly strong performance on complex video-language tasks by modality-constrained baselines with limited access to video-specific information.
2. Adapting Likelihood Selection strategies to video QnA benchmarks for efficient evaluation.
3. Novel VLM-based video QnA framework that extracts concise video specific object-centric information followed by natural language based fusion.

We integrate our MVU framework over multiple different baselines and obtain performance improvements across 20 different datasets establishing both its effectiveness and generality. Our evaluations are performed zero-shot with no video-level training on these datasets which cover video QnA tasks (short, medium, and long videos) as well robotics domain tasks.

5.2 Naive Baselines & Likelihood Selection

In this section, we first establish our problem setting, then discuss adapting likelihood selection for video QnA tasks, and finally introduce two naive LLM based frameworks for video question answering tasks, tagged *Just-LLM* and *Single-Frame-VLM* (see Figure 5.1).

5.2.1 Problem Formulation

We focus on two categories of video understanding tasks:

1. Long Video Question Answering (Multiple-Choice-based Selection)
2. Open Ended Video Question Answering (Text Generation)

For the first task, we construct a unified problem formulation accounting their choice based selection aspect. For the latter, we resort to standard LLM based answer generation.

Consider a video $x_v \in \mathbb{R}^{L\times H\times W\times C}$, a textual question x_t, a set of textual candidate answers $Y = \{y_i, i = 1, ..., M\}$, and a model $V(\cdot)$ selecting one answer from the given

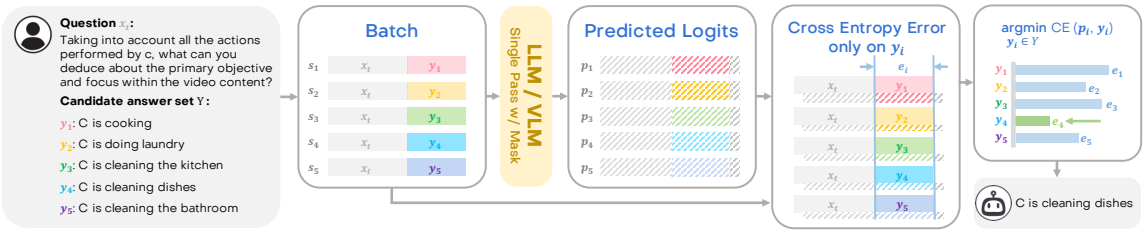

Figure 5.2: **Likelihood Selection Workflow:** We illustrate how the likelihood selection strategy adapted for video QnA tasks can be efficiently parallelized (i.e. calculated with a simple cross-entropy loss in one forward pass, followed by an argmin operation), in contrast to the setting of iteratively generating multiple tokens.

set of answers (noted as $\hat{y} := V(x_v, x_t, Y)$). Selected $\hat{y}$ should ideally be identical to groundtruth y_g. Here L, H, W, C are the number of frames of the video, frame height, width, and number of channels respectively. M is the number of candidate answers. For multiple choice based selection tasks, x_v, x_t, and Y are directly present in dataset. For N-Way Classification tasks, we set x_t as a generic question (details in Section 5.6.1) and formulate Y by applying a fixed template to the labels of all N classes of the dataset. This formulation is used for the remainder of the paper unless a specific exception is noted.

In the case of open-ended video question answering, we follow standard settings of LLM based text generation for video tasks following [160].

5.2.2 Likelihood Selection

The common technique for LLM based approaches tackling question answering (QnA) tasks is likelihood based choice selection (also referred as Cloze Prompting, see [215]). Adopting such likelihood based selection for different tasks (or to VLMs) is however not straightforward [215], leading to most existing long video QnA approaches resorting to LLM based answer generation. In fact, most existing long-video QnA approaches using LLMs / VLMs for choice selection [13, 184, 266] resort to full answer generation followed by embedding or template based matching to ground-truth choices, incurring significant inference costs for evaluation.

In light of this, we explore prompting and likelihood calculation techniques optimal for applying *Likelihood Selection* on long video QnA tasks with either LLMs or even VLMs. Adapting this technique unlocks autoregressive LLMs / VLMs ability to solve multiple selection problems with only *one* forward pass

as illustrated in Figure 5.2. This is in contrast to next token sampling requiring iterative generations dependent on previous outputs for each answer token. This process uses a likelihood measure based on the LLM latent space allowing better semantic awareness compared to exact or template matching. In addition to the candidate answer batching, we follow prior work to include all candidates in the prompt as well. We direct the reader to Table 5.11 for complete details on semantic awareness, candidates in prompts, and video QnA specific implementation.

In addition to the considerable inference speed-up from likelihood selection, we also obtain the additional advantages of avoiding LLM hallucinations and deviations from expected output formats over iterative generation strategies applied to similar visual tasks (see [86]). We empirically validate the improved performance from such behavior in our ablations (see Table 5.6).

5.2.3 Modality Constrained Variants

We next introduce the two modality-constrained variants of our framework tagged *Just-LLM* and *Single-Frame-VLM* (illustrated in Figure 5.1). The former utilizes only the task question injected as language (x_t) with no other task-specific information. Note how this naive variant does not access any information extracted from the video for each task instance. The latter utilizes an additional center visual frame (x_v^c), extracted from the center of the video (x_v) timeline. This variant accesses no *video-specific* data (e.g. temporal or motion information). The center frame usage ensures no temporal information leakage in frame selection for this variant.

We hypothesize that Just-LLM with no access to task-specific knowledge is constrained to generate predictions utilizing its internal world knowledge (e.g. physics, culture, human common sense). We refer to this as *world modality*. For a given question regarding a natural video and a set of candidate answers, there is a possibility that one choice is more probable given how our world operates. In cases that this choice turns out to be correct, the internal world knowledge of the LLM allows it to easily select that choice resulting in above random performance. This variant of our framework highlights such cases in long video QnA tasks. A similar baseline is used in [167].

In the case of Single-Frame-VLM, it is provided with task information but is limited to a single frame, which could possibly provide important scene context. Therein, we refer to this variant as operating with world and *contextual* information modalities. For example, consider a video with a man walking a dog. The scene context of the dog and man combined with the LLM world knowledge and reasoning skills may be sufficient to correctly answer the question with no temporal

Table 5.1: **Modality Constrained Variants:** We report accuracy (%) and inference time per sample (s) on the public subset of EgoSchema (ES-S) and test set of NextQA (NextQA-T) datasets. Note that recent state-of-the-art from [314] (SOTA) and our variants are implemented with common LLMs / VLMs and evaluated under identical settings.

Method	Param	Video Frames	ES-S		NextQA-T	
			Acc	Time	Acc	Time
Random	-	-	20.0	-	20.0	-
Just-LLM	7B	0	45.8	0.41	40.1	0.55
SF-VLM	13B	1	55.8	1.89	51.2	2.03
SOTA	20B	180	50.8	381	54.3	207

or motion information. Performance of this variant highlights the prevalence of similar cases in long video QnA tasks when using LLM based approaches.

We evaluate these two modality-constrained variants and summarize our findings in Table 5.1. We uncover surprisingly strong performance of both variants on two long-video understanding benchmarks. In the case of Just-LLM variant, we achieve performance significantly higher than random selection (+25.8% on ES-S / +20.1% on NextQA-T) using zero visual information. This indicates the large portion of questions in existing video-QnA benchmarks that can be answered correctly purely using world knowledge. We also highlight our Single-Frame-VLM performing on par with state-of-the-art LLM based approach from [314]. In particular, for ES-S we outperform [314] which uses information extracted from 180 frames per video incurring an inference cost over 100 times higher than ours. In light of these findings, we argue that long video understanding approaches in particular must focus on learning information beyond what a single frame baseline can achieve, possibly in an interpretable manner.

Therein, we introduce *Multimodal Video Understanding* (MVU), a simple framework that aggregates multimodal video-relevant information in an interpretable manner using natural language and achieves significant improvements over baselines across multiple datasets.

5.3 Multimodal Video Understanding Framework

In this section, we introduce in detail our Multimodal Video Understanding (MVU) framework that integrates several information modalities extracted from video using *natural language* as a medium for information fusion. Our approach adapts off-the-shelf vision tools to construct a powerful long video understanding agent

that requires no additional training on videos. We first utilize vision tools to extract information relevant to three object-centric modalities from uniformly sampled video frames. Next, we leverage suitable prompt templates to aggregate these as natural language. This video level information is injected into our Single-Frame-VLM variant providing it with video specific awareness. We illustrate an overview of our framework in Figure 5.3.

5.3.1 Vision Tools for Video Analysis

Image trained VLMs contain information valuable for video tasks and have been widely used in prior work [314]. In our proposed framework, we take a step further, exploring more off-the-shelf vision tools trained only on images, in particular object detection and object tracking approaches, in addition to a VLM re-purposed as an image captioner.

We use an image captioner to identify all unique objects present within a video. For this purpose, we prompt a generative vision language model to list all objects within a given video frame (image) in an open-ended manner. We note how a VLM trained only on images is sufficient for this. In our case, we use a VLM identical to the one in [314] but applied on significantly less video frames, making our comparisons fair in terms of model size.

For the case of object detection, we use an open-vocabulary object detector from [168] that is trained only on images, and apply it with object category names from captioner to obtain their location information, i.e. image-space coordinates for each unique object. Given the lightweight nature of this detector in comparison to the image captioner, we note how it can be applied more densely (i.e. on more frames) than the captioner without increasing compute demand significantly. Furthermore, the detector acts as a secondary check, grounding the object category names to individual frames, and therein countering any object hallucinations by the captioner.

Our final tool is an object tracker from [260] used to convert our per-frame object detections into motion trajectories spread across the entire video. We feed the tracking algorithm with the locations of each object alongside per-object features extracted from our detector in order to construct motion trajectories for each unique object.

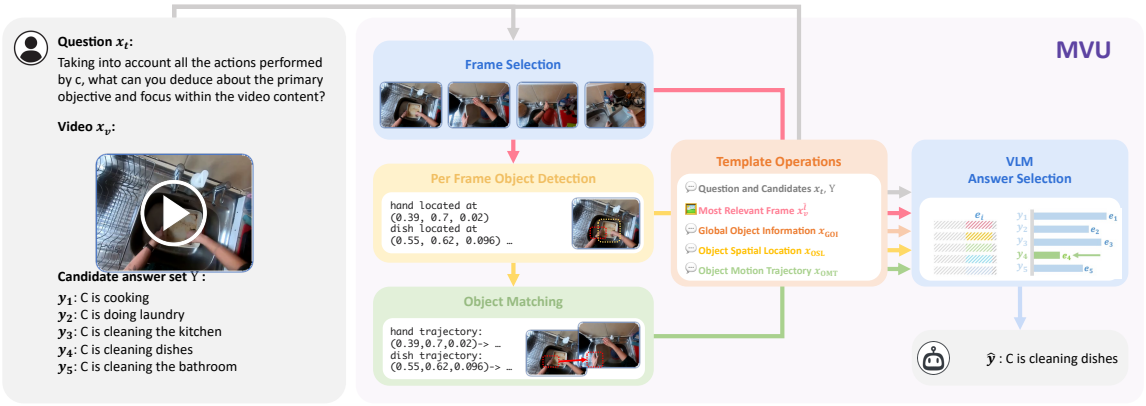

Figure 5.3: Overview of proposed framework for Multimodal Video Understanding, MVU.

5.3.2 Object-Centric Information Modalities

Given off-the-shelf tools suitable for extracting information from videos, we next focus on the exact forms of information, i.e. three object-centric information modalities. We consider all object categories across the video, spatial locations of individual object instances, and their movement across time. We define these as follows:

1. **Global Object Information (x_{GOI}):** In this stage, we introduce global information that spans beyond a single video frame. For a given video, we first uniformly sample 8 frames. For each of the 8 selected frames, we utilize our image captioner to generate object lists and obtain a set of distinct object categories contained within each frame across the video.

2. **Object Spatial Location (x_{OSL}):** Given objects present per video, we utilize our open-vocabulary object detector to localize each object category (from previous stage) on to frame coordinates. Categories not localized by the detector are dropped. Additionally, we utilize similarity of feature vectors for same class objects to track object instances across frames using our tracker. Following prior work [206], we calculate average center coordinates and scale value for each object instance across all frames. This results in a set of distinct objects O across the video, $O = \{(o_1, q_1), (o_2, q_2), ...\}$. Here, o_k describes the object category in natural language while q_k contains the x, y coordinates of object center and the scale term (area of minimal object bounding box as a ratio to image size, i.e. box area ÷ image size).

3. **Object Motion Trajectory (x_{OMT}):** Next, we leverage the calculated cross-frame object tracks and compute motion trajectories for each object. This modifies our set of distinct objects, pairing each object o_k with its trajectory $(o_k^1 \rightarrow o_k^2 \rightarrow ...)$ across the video frames. We construct an updated set $Z = \{(o_1, q_1^1 \rightarrow q_1^2 \rightarrow ...), (o_2, q_2^1 \rightarrow q_2^2 \rightarrow ...), ...\}$. Intuitively, this information should explicitly capture object motion information.

We provide further details including examples of each information modality for selected samples (video question pairs) in Section 5.6.1.

This pipeline for extracting per-frame information using an image-trained VLM closely resembles prior work such as [314]. While motivated by such work, we explore the direction of how more fine-grained information could be extracted from videos to solve these tasks more efficiently. Given the role of object interactions in defining the various actions and events in videos, we hypothesize that extracting object-centric information (as opposed to generic frame-level descriptions) followed by modeling of their temporal dependencies would provide more concise representations better suited to efficiently solve these tasks.

5.3.3 Language based Fusion

Inspired by [310], we construct our overall framework by injecting these three forms of object-centric information into our setup using natural language. We represent each modality in a fixed template-based fusion. Global object information is represented as a list of category labels, e.g., x_{GOI} = {*person*, *oven*, *dishwasher*, ..., *sink*}. Object spatial location modifies this list to include center coordinates (x, y) and scale (s) where scale is the area percentage occupied by the best-fitting object bounding box. For e.g., x_{OSL} = {*person located at* (0.2, 0.3, 0.07), ... , *oven located at* (0.8, 0.6, 0.04)}. Finally, object motion trajectories update the list to contain frame-level trajectories, e.g., x_{OMT} = {*person moving as* [0.2, 0.3, 0.07] $\rightarrow$ [0.2, 0.4, 0.06] $\rightarrow$ [0.2, 0.6, 0.08], *oven moving as* ... }. Similar to the examples, information from each object-centric modality is represented in textual form to allow their direct fusion and integration into our framework (as additional language inputs). Therein, we describe the resulting setup, our overall framework MVU as follows,

$$\hat{y} = \mathcal{F}_{\text{MVU}}(x_t, x_v^c, x_{\text{GOI}}, x_{\text{OSL}}, x_{\text{OMT}}) \tag{5.1}$$

where x_v^c is the center frame extracted from the video x_v (more details in Section 5.6.1). In comparison to prior work such as [314], we note that our fused

information is more concise allowing better utilization of the fixed context length in an LLM (see Section 5.6.14 for more details).

5.4 Experiments

In this section, we first discuss our experimental setup and datasets. Next, we evaluate MVU on multiple video question-answering and robotics task benchmarks followed by ablative studies.

Experimental Setup: Our proposed MVU framework and its variants use off-the-shelf models trained on images, thus requiring no re-training of these models. For our evaluations, we directly use these models, utilizing two NVIDIA RTX A5000 24GB GPUs for inference. We evaluate on two video question answering datasets focused on long-form videos: EgoSchema [162] and NExT-QA [276]. We also evaluate using a series of robotics datasets from the Open X-Embodiment robotics dataset [179] to test our model generality (more details in Section 5.4.2). We discuss further details of pretrained models and datasets in Section 5.6.2. Also, note that none of the pretrained components of our framework undergo any form of video-level training.

5.4.1 Long Video Question Answering

Long video question answering benchmarks aim to measure causal and temporal reasoning abilities of models over long temporal windows [162, 276]. In this section, we evaluate our framework on two benchmark datasets and present our results in Table 5.2 and Table 5.3.

On EgoSchema dataset, results reported in Table 5.2 demonstrate the state-of-the-art performance of our framework. We integrate MVU over SF-VLM and LVNet [186] baselines for fair comparison to work operating under different settings. We reiterate how our approach is both zero-shot and requires no video-level training (and our selected baselines are similar). In comparison to prior work utilizing open models, our `SF-VLM+MVU` achieves clear performance improvements, even out-performing works using video-caption supervision for training [13, 184]. Compared to methods utilizing proprietary closed language models extending to trillion parameter scale [167, 261, 264, 314], our `LVNet+MVU` variant using similar scale achieves improved performance. We also implement several such large-scale approaches under scaled-down common settings as our smaller variant (details in Section 5.6.3), where we again achieve clear performance gains.

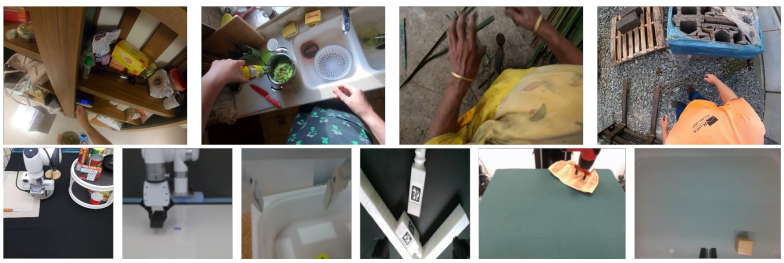

Figure 5.4: **Data Visualization:** Example video frames from EgoSchema (top) vs OpenX (bottom) datasets. Robotics domain videos (bottom) appear out of distribution given their controlled environment and robot movements.

Next, we evaluate our framework on the NextQA benchmark and report these results in Table 5.3. We similarly integrate MVU with two baselines. Our MVU achieves state-of-the-art results under zero-shot settings. While [301] outperforms our approach, we note how they require video-caption localization pretraining and appears to overfit to this dataset considering their relatively lower performance on other datasets (see Table 5.2).

We also evaluate MVU on the LongVideoBench dataset which contains even longer videos and present these results in Section 5.6.11. While these three datasets focus on MCQ style QnA, we also explore the generality of our MVU framework on open-ended style QnA tasks in Section 5.6.10.

5.4.2 Robotics Domain Action Recognition

We investigate generalization capabilities of our proposed MVU by evaluating across datasets from robotics domain Open X-Embodiment [179], following a QnA style formulation of the dataset (details in Section 5.6.4). We highlight visual differences of this data in Figure 5.4. We present evaluations in Table 5.4, which indicate clear performance improvements for MVU over the baseline from [314]. The purpose of this experiment is to evaluate the generality of our approach to video domains different from everyday natural videos. We take these promising results to indicate the strong generality of our framework. Furthermore, we note how our modality constrained variants do not perform significantly better than random on these robotics domain tasks (details in Section 5.6.5). We attribute this to the significant domain shift in terms of the world of operation in this domain (i.e. robotics tasks tend to involve controlled environments very different to what humans face on an everyday basis).

5.4.3 Ablations

In this section, we systematically dissect our overall MVU framework to establish the usefulness of each of its individual component (see Section 5.6.12 for more ablations). We first ablate our three different information modalities and report these results in Table 5.5. Our evaluations indicate clear performance improvements from each of our object-centric information modalities.

We next perform ablations on our adaptation of likelihood selection strategy for video QnA tasks using Ego-Schema subset (ES-S) and Next-QA test-set (NQA-T) . These results reported in Table 5.6 indicate clear performance boosts due to our adaptation of likelihood selection (LS). When removing LS, standard generation (i.e. generate an answer and match against ground-truth selection following [314]) is utilized with our MVU framework. We also report naive adaptation of LS following [215] where the choice options are directly used, highlighting the importance of our prompting techniques. We also note the accuracy gains obtained through LS, and attribute these to reduced LLM output hallucination and deviations from expected output formats, that are commonly observed with iterative generation [86].

We next ablate our overall framework against the existing work, [314], by replacing our MVU object-centric information pipeline with the frame description approach in [314]. We construct a setup identical to our framework except for the inputs to our final stage VLM replaced with frame level descriptions. These results reported in Table 5.7 indicate the clear significance and improvement of our proposed object-centric information pipeline over simple frame descriptions. The 8 frame variant is the same speed comparison as MVU uses captioner only on 8 frames. Our MVU outperforms both that and the slower 16 frame baseline. We also note the performance drop in the baseline when increasing the number of frames from 16 to 180. While consistent with observations in prior works for long-video tasks [162], we attribute this drop to decreased signal-to-noise ratio with the introduction of additional frame descriptions. This further highlights the importance of selecting useful information from video frames, and we reiterate how the object-centric information in our MVU framework serves this purpose.

5.5 Conclusion

In this work, we present a multimodal video understanding framework, termed MVU, that achieves state-of-the-art performance on complex video understanding tasks. In particular, evaluations on long-video question answering and robotics

domain question answering demonstrate the strong performance of our MVU framework as well as its generality. We also adapt likelihood selection for efficient LLM-based answer choice selection, separate video-specific information into three object-centric modalities, demonstrate automated extraction of such information using off-the-shelf vision tools, and propose language-based fusion of this multimodal information.

We also presented two modality-constrained baselines that uncover surprising insights relevant to LLM based video QnA which serves as a basis for our subsequent MVU framework. Furthermore, these results highlight the need for careful evaluation of LLM-based video QnA approaches. Revisiting our original motivation on "*what we can learn from videos, beyond scene context understood from a single natural image*", in this work our two modality-constrained variants uncover surprising insights relevant to this question. We first achieve strong results on long-video understanding benchmarks using no video-specific data, and build over that baseline to showcase the additional performance gains achievable through injecting video-specific information.

5.6 Additional Details

5.6.1 Prompting and Template Operations

In Section 5.3.2 and Section 5.3.3, we utilize 3 distinct prompts and fusion templates for generating joint textual inputs to be processed by the LLM. The 3 distinct prompt categories correspond to our Global Object Information (x_{GOI}), Object Spatial Location (x_{OSL}), and Object Motion Trajectory (x_{OMT}) modalities. We first describe our exact templates as Python pseudo-code in Table 5.8.

The above templates depend on each of the modalities represented in textual form (i.e. `GOI_data`, `OSL_data`, `OMT_data`). We describe their exact textual forms next using examples in Table 5.9.

In this example (for a single video), the `GOI_data` list contains 11 distinct object categories discovered across all 8 selected frames for this video. In `OSL_data`, this category list is grounded to each frame using our object detector. We apply this on 16 uniformly sampled frames as opposed to only 8 used with the captioner. While this stage removes some categories (which we assume could be object hallucinations [206]), it also identifies categories at the instance level. We draw attention to the two different instances of a dish in our `OSL_data` for this example. Also, note that the single spatial coordinate reflects the average location of that object across all 16 (or the number of frames it is detected in) following the setup in [206]. Our object

tracks calculated across frames are utilized for this averaging (i.e. distinguish the two dishes). For our OMT_data, we again leverage our tracks where each object instance is matched across frames and construct explicit sequences of object locations across frames. While ignoring the actual frame indexes, we only consider the object trajectory using frames where they are detected. Note that an object trajectory could be limited to a single location or a variable number of multiple locations. Also, for these trajectories, we introduce an additional scale factor for each object location. This scale factor is the ratio of the object bounding box area to image area, i.e. $(\texttt{obj\_width} * \texttt{obj\_height}) \div \texttt{im\_size}$. This is introduced with an aim of possibly providing some level of depth information. In terms of generating object tracks, we utilize intermediate features from our object detector and perform feature matching based tracking.

5.6.2 Details on Pretrained Models and Datasets

We describe in detail the pretrained models used to construct our framework as well as the multiple datasets used in evaluating our framework.

Models: Our framework utilizes three distinct off-the-shelf models for its various operations, namely *a*) an LLM / VLM for likelihood selection, *b*) a generative VLM for extracting object list from a frame, and *c*) an open-vocabulary detector for object localization. We use `LLaVA-v1.5-13B` [149] for likelihood selection and frame object list generation. For object localization, we use `OWL-ViT-B/32` [168]. Unless explicitly specified, we use the above setup in all our experiments. Variants of our framework uses LLMs `Llama-2-7b-Chat`, `Gemma-7b-IT`, and `Mistral-7B-Instruct` (default) for likelihood selection. Apart from these off-the-shelf models, our framework involves zero additional training. We also reiterate that no components of our framework undergo any form of video-level training.

Datasets: We use multiple video datasets for evaluation under question-answering or n-way classification settings. For video question answering, we select two datasets focused on long-form videos: EgoSchema [162], NExT-QA [276]. EgoSchema is a long-form ego-centric video question-answering benchmark, consisting of a 500-video public subset (EgoSchema-S) and a full 5000+ video evaluation set (EgoSchema-F) accessed only through evaluation servers. This dataset spans over 250 hours and is specially constructed to ensure that *questions require awareness of a longer temporal window for correctly answering* [162]. Example images of EgoSchema are shown in Figure 5.4. NExT-QA similarly contains long-form videos with a focus on requiring causal & temporal action reasoning as well as common scene comprehension for correctly answering. It contains a validation set

(NExT-QA-V) of 4996 video-questions pairs and a test set (NExT-QA-T) of 8564 video-question pairs. We also use a series of robotics datasets from the Open X-Embodiment robotics dataset [179] for video question answering in a different domain (more detail in Section 5.4.2). In only one of our ablations aimed at analyzing the motion understanding aspect of our framework, we utilize a fine-grained action recognition dataset, Something-Something-v2 [77], that contains 174 action categories focusing on object motions by replacing object category nouns with '*something*' in textual descriptions of each action category.

5.6.3 Details on Baselines

In Section 5.4.1, we evaluate performance on long-video understanding tasks using work in [314] and [261] as two baselines for comparison. However, both these methods utilize closed-source, proprietary LLMs (i.e. GPT-4) with parameter counts on the order of trillions (over 100X our model size) deeming their direct comparisons unfair. In the case of [314], we replicate their method (using their open-source repository and pre-trained models following [109]) utilizing an open-source LLM of comparable parameter count as our framework. For [261], the authors directly report results for a variant with a similar parameter count as ours. We utilize these evaluations as our point of comparison.

We also replicate prior work, LVNet [186], that exhibits state-of-the-art results. For this, we use their official code (`https://github.com/jongwoopark7978/LVNet`) and integrate our MVU framework over this baseline.

We highlight that re-implementations of these baselines utilize common LLMs / VLMs as our MVU framework followed by identical evaluation protocols to ensure fair comparison.

5.6.4 Robotics Domain Dataset Details

The Open X-Embodiment dataset is an extensive collection of visuomotor robotics datasets, encompassing a wide range of tasks and environments. It is designed to facilitate research in visuomotor control, providing rich sensory inputs and motor outputs for training and testing robotic systems. However, the videos are usually taken in a controlled environment and they do not always contain meaningful objects, which makes the samples in the dataset usually out of general video distribution (See Figure 5.4).

For our analysis, we specifically select datasets within this collection that contain expert episodes accompanied by corresponding language instructions and

adapt them into video classification datasets. We treat each trajectory within the dataset as a video clip, with its associated language instruction serving as the video caption (classification label). For each episode, the model is tasked with identifying the correct language instruction from a set of five options, considering a video clip uniformly downsampled to 8 frames. The incorrect options are randomly chosen from the dataset to ensure a diverse and challenging selection. In instances where the datasets have multiple cameras for each episode, we treat the videos captured by each camera as distinct datasets.

5.6.5 Discussion on Modality Constrained Evaluation

We evaluate the two modality-constrained variants of our approach, Just-LLM and Single-Frame-VLM (details in Section 5.2.3) and summarize these findings in Table 5.1. We uncover surprisingly strong performance of both variants on two long-video understanding benchmarks. Note how these approaches use no video-specific information to generate predictions.

We highlight how our best Just-LLM variant achieves performance significantly higher than random selection (+25.8% on EgoSchema-S / +20.1% on NextQA-T) using zero visual information. This indicates the large portion of questions in existing video-QnA benchmarks that can be answered correctly purely using world knowledge. We also highlight our single frame variant performing on par with some existing state-of-the-art (gray). In particular, for EgoSchema-S we outperform [314] which uses information extracted from 180 frames per video incurring an inference cost over 100 times higher than ours. In light of these findings, we argue that long video understanding approaches in particular must focus on learning information beyond what a single frame baseline can achieve.

We also evaluate these same modality-constrained variants on robotics domains tasks and report these results in Table 5.10. In contrast to the results on standard long-video QnA benchmarks, the robotics domains results are more aligned with intuition: the no-visual input Just-LLM performs on par with random and the Single-Frame-VLM marginally outperforms random selection.

We attribute this difference in performance to the nature of robotics domain tasks. They tend to involve controlled environments with often naive, meaningless tasks purely for robot testing purposes. These may not necessarily align with human commonsense or other constraints dependent on knowledge of our world. Therein, the clear ability of LLMs to solve general everyday video tasks (e.g. EgoSchema, NextQA performance in Table 5.1) using its world knowledge may not be applicable to robotics domain tasks. Utilizing different domain benchmarks, in

particular robotics tasks, provides a much more representative evaluation of LLM based video QnA approaches.

5.6.6 Likelihood Selection

In this section, we present the prompts and templates used to adapt likelihood selection inference [215] to our video QnA tasks. Our experimentation shows significantly higher sensitivity (to prompt style) of LLM performance on QnA tasks when using like likelihood selection in comparison to sequential text generation (consistent with findings in [215]). We evaluate a series of different prompt templates on the EgoSchema and Next-QA dataset to discover optimal combinations. The best prompt templates used in our final framework are presented in Table 5.11 as Python pseudo-code. For Next-QA in particular, the average zero-shot accuracy could vary from 35% to 55% with slight variations of the prompt templates.

Our optimal prompt templates for the standard video QnA tasks also generalized to our robotics domain QnA tasks. Nevertheless, we highlight the possibility of needing some prompt template tuning when applying our framework to different domains. We also note that while our prompt selection process was guided by heuristics and experimentation, there may be other similar prompts performing equally well or surpassing our optimal selection.

5.6.7 Implementation Details

We revisit the generation process of autoregressive LLMs and their visual extensions (VLMs). They commonly use iterative prediction of next tokens conditioned on prior outputs to generate complete natural language outputs. Such a generation process is usually modeled as sampling from a conditional likelihood shown as Equation (5.2), where $\hat{y}^j$ stands for the j^{th} token in a textual sequence $\hat{y}$ autoregressively generated by the model.

$$P(\hat{y}|x_t) = \prod_j P(\hat{y}^j|\hat{y}^{1,\ldots,j-1}, x_t) \tag{5.2}$$

The dependency on prior output $\hat{y}^{1,\ldots,j-1}$ makes this process both computationally costly and redundant in the case of choice-based answer selection tasks. Alternately, given the closed set of Y in choice-based selections tasks, we formulate $P(y_i|x_t)$ for any $y_i \in Y$ with no dependency on any model generated output ($\hat{y}$) as,

$$P(y_i|x_t) = \prod_j P(y_i^j|y_i^{1,\ldots,j-1}, x_t) \tag{5.3}$$

Assume a perfect LLM, intuitively when y_i is a proper answer to the question x_t (say $y_i = y_g$), the conditional likelihood $P(y_i|x_t)$ should be larger than any other $P(y_w|x_t)$ where y_w is a wrong answer to question x_t. In fact, modern LLMs are trained with a similar objective [194]. Motivated by this standard LLM training objective, we estimate the relative numerical scales of conditional likelihood on different answers $P(y_i|x_t)$ using a cross-entropy error e_i, given their equivalence (negative log-likelihood and multiclass cross-entropy, see Section 4.3.4 in [22]). We calculate e_i with a single forward pass of LLM without detokenization and the selection can be made by simply picking up the answer with the lowest error, equivalent to the highest conditional likelihood among all the answers.

This sets the ground for *Likelihood Selection*, also referred to as Cloze Promting in [215], first illustrated with a toy example in Figure 5.2, where the task is vanilla question-answering with only textual context and the model takes one question x_t as well as $M = 5$ candidate answers $y_{1,\dots,5}$. To find the best answer, we simply concatenate the question with each candidate independently ($s_i = \text{concat}\,(x_t, y_i)$) and pad them into a batch $\{s_{1,\dots,5}\}$. Then the LLM takes the batch of five samples with causal attention masks and performs one inference forward pass, resulting in five shifted logits $\{p_{1,\dots,5}\}$. Next, we shift the input sequence s_i to align the logits p_i and calculate the average cross-entropy error only on tokens of y_i Finally, the answer with the smallest e_i will be picked up as the best answer. The method can be formulated as in Equation (5.5) using equivalence of negative log-likelihood to cross-entropy in Equation (5.4). Here n_i stands for the token sequence length of y_i and p_i^j stands for logits of the j^{th} token in $p_i = V(\text{concat}\,(x_t, y_i))$ with logits limited to only those of y_i.

$$e_i(y_i) = \text{CE}(p_i, y_i) = \frac{1}{n_i} \sum_j^{n_i} \left(\text{CE}(p_i^j, y_i^j) \right) \approx \sum_j^{n_i} -\log P(y_i|x_t) \tag{5.4}$$

$$\mathcal{F}_{\text{LS}}(Y, x_t) = \arg\max_{y_i \in Y} P(y_i|x_t) = \arg\min_{y_i \in Y} e_i(y_i) \tag{5.5}$$

In summary, Likelihood Selection performs one single forward pass to extract the network logit outputs, calculates error (e_i) on each choice, and selects the choice with the lowest error. Note that our method does not utilize any iterative autoregressive generation using the LLM. This results in considerable speed-ups for inference time. We also obtain the additional advantages of avoiding LLM hallucinations and deviations from expected output formats over iterative generation strategies applied to similar visual tasks [86] leading to better accuracy (see Tab. 6.). In Section 5.2.3, we demonstrate both our speed gains and performance

improvements.

Furthermore, Likelihood Selection is a generic method that can be easily extended to autoregressive VLMs, and in principle, there is no reason it could not also be used with extra modalities besides language. We validate this across all our experiments using the multimodal MVU framework.

5.6.8 Distinction from Exact Match

As described in the previous section, likelihood selection uses a likelihood measure which is the likelihood (probability) of the model generating the given sentence (as opposed to being an exact match). This likelihood measure is also used as the training loss when training LLMs. Given how LLMs trained with this loss (i.e. all decoder based LLMs such as LLaMA, Gemini, GPT) are highly effective at handling semantic meaning, it follows that this loss can capture semantic meaning. This likelihood measure is calculated within the LLM latent space. This is equivalent to the probability (or likelihood) of that answer being generated by the LLM conditioned on the input question. We derive this in detail in Appendix F. Relating to the same example, this means that likelihood is an estimate of how likely the model would predict 'C is washing plates' as opposed to making that exact match. This means predictions closer to the target such as 'C is cleaning dishes' would also gain high likelihood values.

In fact, we validate this second point through a toy example. We provide an LLM with the question "X is cleaning dishes in the kitchen. What is X doing? a) washing plates, b) cleaning laundry, c) painting dishes. The correct choice is:" and calculate the likelihood for each of the 3 responses. The calculated likelihoods are 0.996, 0.006, 0.007 for a, b, c respectively (highest is selected), despite response (a) having no common words with the original statement unlike (b) and (c). This illustrates the ability of likelihood selection to capture semantic meanings.

5.6.9 Detailed Prompting Example

We also note that while different choices are repeated along the batch, our likelihood implementation actually follows prior approaches where all answer candidates are fed together to the language model in addition to organizing the Q-A pairs in a batch dimension. Taking one simplified toy example, given a question "Where is the dog?" and answers "mat, table, bench", we use three queries along batch dimension as:

- `Where is the dog? Select the correct response from: a) mat, b) table, c) bench. The correct response is a) mat.`
- `Where is the dog? Select the correct response from: a) mat, b) table, c) bench. The correct response is b) table.`
- `Where is the dog? Select the correct response from: a) mat, b) table, c) bench. The correct response is c) bench.`

In fact, applying likelihood selection without such prompting leads to significantly low performance for some datasets. We show this in Table 5.6 which we repeat here as Table 5.12.

5.6.10 Open-Ended Video Question Answering

In this section, we explore the ability of our proposed MVU framework to operate on open-ended video question answering (QnA) tasks. For this purpose, we evaluate on the Activity-Net dataset [303] reporting the accuracy metric. We follow evaluation settings identical to [160] for these evaluations.

Given the nature of open-ended QnA tasks (i.e. no answer choices, generate free form answers), we use standard generation instead of likelihood selection. We match the generated answers against ground-truth following [160]. We present these results in Table 5.13 where our MVU achieves strong results and clear improvements over the similar LLM based approach from [314]. We compare against multiple recent approaches that use similar capacity LLMs VLMs for open-ended video QnA. We take these results as another indication to the generality of our MVU framework on video QnA tasks beyond MCQ style.

5.6.11 Longer Video Question Answering

While established long video benchmarks used as the key evaluations in numerous prior work [109, 167, 186, 264, 267, 269, 314] limit to roughly 1-3 minute long videos, some newer datasets include even longer videos [273]. We explore such even longer videos by evaluating our method on the LongVideoBench dataset [273].

We select Phi-3-Vision-Instruct [1] as our baseline since it is the best performing model we can replicate within our compute budget. We note that larger sized models using significantly larger context lengths are difficult to replicate within academic compute restraints. Results using this baseline from [1] and our MVU

framework integrated over it are presented in Table 5.14. MVU gains clear performance gains in this longer video dataset.

5.6.12 Additional Ablations

In this section, we repeat part of our ablation from Table 5.5 focused on the object motion trajectory modality inputs. We note that common video QnA benchmarks require minimal understanding of object motion to answer most questions. Our goal is to explore the value of motion information in a more relevant tasks.

Therein we investigate a new motion focused dataset, Something-Something-v2 [77] (SSv2), only for this single ablation. The SSv2 dataset focuses on motion-based category discrimination, providing an ideal evaluation to measure the usefulness of our object motion trajectory modality. We benchmark on a subset of this dataset following [145] and report these results in Table 5.15. Our results while exceeding their performance also indicate the clear performance gains obtained when injecting the object motion trajectory modality into our MVU framework.

We also provide an ablation on frames used with our MVU framework in Table 5.16. Increasing the number of frames leads to improved performance in contrast to some prior works [162] highlighting how our information fusion pipeline allows better utilization of the LLM context length. Additionally, the lightweight object detector and tracker used in MVU allows scaling the number of frames with a lesser increase in inference time.

5.6.13 Tokenization in LLMs

Most modern LLMs utilize Byte-Pair Encoding (BPE) tokenization [223] to convert natural language into discrete tokens that are mapped to vocabulary embeddings. This process is learned from language itself and the resulting tokenization may sometimes break complete words into pieces (e.g. `example` $\rightarrow$ `ex-am-ple`). Given our utilization of logits extracted from an LLM forward pass, we note that each logit corresponds to a single token, which may at times be the embedding of some meaningless word piece. However, our calculation of a joint likelihood across a sequence of tokens ensures a meaningful process, which is validated by our strong results.

5.6.14 LLM Context Length

Using LLMs for long video understanding has proven successful [269, 314] but handling long context lengths remains a key issue [162, 186], often leading to lower performance when additional frame information is provided to the LLM. This draws importance to frame selection, but we argue that alternate forms of information bottlenecks can also provide improvements, often complementary to frame selection.

In MVU, instead of naively collecting all information within a frame, we only collect object centric spatial and motion information, allowing to process more frames at a fixed context length. In other words, MVU information extraction from multiple frames can be viewed as an alternative to frame selection. This is because our object information extraction indirectly acts as an information bottleneck similar to frame selection. For frames without objects of interest, no information is extracted. For multiple frames containing the same object (identified by our object tracker), the repetitive information is removed. This resembles the idea of selecting useful information from multiple frames.

In fact, when comparing the average token length for a similarly performing baseline (implemented under identical settings using a common LLM), we use less tokens (context length) to achieve similar results. We show these results in Table 5.17.

5.6.15 Additional Baselines

We implement a multi-frame baseline directly using LLaVA-1.5 [149] with no video level training. These results are reported in Table 5.18. Results indicate that directly adding multiple frames to a VLM with no video level training does not lead to improved performance. Similar trends are observed in prior work [109]. These findings highlight the importance of careful per-frame information extraction and cross frame information fusion proposed in our MVU.

Table 5.2: **Ego-Schema Dataset Evaluation**: We report top-1 accuracy (%) for video question answering on Ego-Schema [162] test set (5031 videos). Our proposed MVU achieves state-of-the-art performance on this benchmark under *zero-shot operation with no video level training*. We also draw attention to our modality-constrained SF-VLM baseline that achieves surprisingly competitive performance.

Method	Zero Shot	Video Training	Closed Model	Params	Full
Random Selection	-	-	-	-	20.0
VIOLET [67]	✓	✓	✗	198M	19.9
FrozenBiLM [292]	✓	✓	✗	1.2B	26.9
SeViLA [301]	✓	✓	✗	4B	22.7
mPLUG-Owl [296]	✓	✓	✗	7.2B	31.1
InternVideo [266]	✓	✓	✗	478M	32.1
ImageViT [184]	✗	✓	✗	1B	30.9
SeViLA+ShortViViT [184]	✗	✓	✗	5B	31.3
LongViViT [184]	✗	✓	✗	1B	33.3
MC-ViT-L [13]	✗	✓	✗	424M	44.4
InternVideo2 [268]	✓	✓	✗	7B	55.8
Tarsier [256]	✓	✓	✗	7B	49.9
Tarsier [256]	✓	✓	✗	34B	61.7
Vamos [261]	✓	✗	✗	13B	36.7
LLoVi [314]	✓	✗	✗	13B	33.5
LangRepo [109]	✓	✗	✗	12B	41.2
Vamos [261]	✓	✗	✓	1.8T	48.3
LLoVi [314]	✓	✗	✓	1.8T	50.3
LifelongMemory [267]	✓	✗	✓	1.8T	62.4
MoreVQA [167]	✓	✗	✓	-	51.7
VideoAgent [264]	✓	✗	✓	1.8T	54.1
VideoTree [269]	✓	✗	✓	1.8T	61.1
LVNet [186]	✓	✗	✓	1.8T	61.1
SF-VLM (ours)	✓	✗	✗	13B	36.4
SF-VLM + MVU (ours)	✓	✗	✗	13B	37.6
LVNet + MVU (ours)	✓	✗	✓	1.8T	61.3

Table 5.3: **Next-QA Dataset Evaluation**: We report top-1 accuracy (%) for the Next-QA dataset [276]. Our proposed MVU achieves state-of-the-art results under zero-shot settings with *no video-level training*. In table header, ZS stands for zero-shot and VT stands for video level training.

Method	ZS	VT	Params	Cau.	Tem.	Des.	All
Random Selection	-	-		20.0	20.0	20.0	20.0
CoVGT [279]	✗	✓	149M	58.8	57.4	69.3	60.0
SeViT [115]	✗	✓	215M	-	-	-	60.6
HiTeA [295]	✗	✓	297M	62.4	58.3	75.6	63.1
InternVideo [266]	✗	✓	478M	62.5	58.5	75.8	63.2
MC-ViT-L [13]	✗	✓	424M	-	-	-	65.0
BLIP-2 [133]	✗	✓	4B	70.1	65.2	80.1	70.1
SeViLA [301]	✗	✓	4B	74.2	69.4	81.3	73.8
LLama-VQA-7B [117]	✗	✓	7B	72.7	69.2	75.8	72.0
Vamos [261]	✗	✓	7B	72.6	69.6	78.0	72.5
Just-Ask [291]	✓	✓	66M	31.8	30.4	36.0	38.4
VFC [170]	✓	✓	164M	45.4	51.6	64.1	51.5
InternVideo [266]	✓	✓	478M	43.4	48.0	65.1	49.1
SeViLA[301]	✓	✓	4B	61.3	61.5	75.6	63.6
CaKE-LM [237]	✓	✗	2.7B	35.7	35.3	36.8	34.9
LLoVi [314]	✓	✗	13B	55.6	47.9	63.2	54.3
ViperGPT [241]	✓	✗	175B	-	-	-	60.0
LLoVi [314] (GPT-4)	✓	✗	1.8T	69.5	61.0	75.6	67.7
MoreVQA [167]	✓	✗	1.7T	70.2	64.6	-	69.2
VideoAgent [264]	✓	✗	1.7T	72.7	64.5	81.1	71.3
VideoTree [269]	✓	✗	1.7T	75.2	67.0	81.3	73.5
LVNet [186]	✓	✗	1.8T	75.0	65.5	81.5	72.9
SF-VLM + MVU (ours)	✓	✗	13B	55.7	48.2	64.2	55.4
LVNet + MVU (ours)	✓	✗	1.8T	75.2	66.8	81.3	73.3

Table 5.4: **OpenX Detailed Results:** We report accuracy (%) for the VideoQA formulation of Open X-Embodiment benchmark. MVU achieves clear improvements over random selection and LLoVi baseline [314]. In table header, Obs. (observation), size, CC (class count) stand for camera used, number of videos, and number of unique language instructions per dataset, respectively. In observation column, T stands for third-person view (stationary camera that does not move with robot), while F denotes first-person view where camera is mounted on moving robot. Note that total is average weighted by dataset size.

Dataset	Obs.	Size	CC	Random	Baseline	MVU
ASU TableTop Manipulation	T	110	83	13.6	19.1	**20.9**
Berkeley MVP Data	F	480	6	20	26.0	**33.1**
Berkeley RPT Data	F	908	4	24.6	23.1	**26.2**
CMU Play Fusion	T	576	44	20.3	34.0	**35.6**
CMU Stretch	T	135	5	23	18.5	**24.4**
Furniture Bench	T	5100	9	20.2	24.8	**26.4**
Furniture Bench	F	5100	9	20.2	22.6	**24.9**
CMU Franka Pick-Insert Data	T	631	7	18.7	19.3	**21.2**
CMU Franka Pick-Insert Data	F	631	7	23.1	**57.8**	49.3
Imperial F Cam	T	170	17	20	22.9	**24.1**
Imperial F Cam	F	170	17	23.5	20.6	**24.7**
USC Jaco Play	T	1085	89	21.8	26.4	**30.6**
USC Jaco Play	F	1085	89	19.4	28.6	**32.4**
NYU ROT	T	14	12	21.4	**57.1**	**57.1**
Roboturk	T	1959	3	34.7	43.0	**44.2**
Stanford HYDRA	T	570	3	35.1	54.7	**68.2**
Stanford HYDRA	F	570	3	31.2	45.3	**48.9**
Freiburg Franka Play	F	3603	406	20.4	**32.2**	31.6
Freiburg Franka Play	T	3603	406	19.7	21.8	**24.0**
LSMO Dataset	T	50	2	34.0	68.0	**72.0**
UCSD Kitchen	T	150	8	19.3	32.0	**32.7**
Austin VIOLA	T	150	3	26.7	32.7	**33.3**
Austin VIOLA	F	150	3	30.0	33.3	**34.0**
Total	-	27000	-	22.1	28.5	**30.4**

Table 5.5: **MVU Ablation:** We report accuracy (%) on public subset of EgoSchema (ES-S). In table header, VI stand for visual inputs and GOI, OSL, OMT refer to our object-centric information modalities (see Section 5.3.2). Each information modality results in clear performance improvements with our full MVU achieving best performance.

Method	VI	GOI	OSL	OMT	ES-S
Just-LLM (ours)	✗	✗	✗	✗	45.8
SF-VLM (ours)	✓	✗	✗	✗	55.8
MVU (ours)	✓	✓	✗	✗	56.4
MVU (ours)	✓	✓	✓	✗	58.6
MVU (ours-full)	✓	✓	✓	✓	**60.3**

Table 5.6: **Likelihood Selection (LS) Ablation:** Results indicate clear improvements in both accuracy (%) and inference time (s) with our adaptation of likelihood selection for video tasks.

Method	LS	ES-S	NQA-T	Time
Generation	✗	56.4	55.3	12.7
LS-Naive	✗	58.2	35.8	2.42
LS-MVU (ours)	✓	**60.3**	**55.4**	**2.42**

Table 5.7: **Baseline Ablation:** We replace information input to final stage VLM with frame descriptions following [314]. Accuracy (%) on public subset of EgoSchema (ES-S). Time in seconds (s).

Method	Frames	ES-S	Time
Baseline	180	55.4	207
Baseline	8	55.8	2.38
Baseline	16	56.2	4.72
MVU (ours)	16	**60.3**	2.42

Global Object Information (x_{GOI})

```
"Consider following objects in video to answer the question:" + \
", ".join(GOI_data) + ".  " + task_question
```

Object Spatial Location (x_{OSL})

```
"Consider following objects with spatial location as
(x, y, area) coordinates in video to answer the question:" + \
", ".join(OSL_data) + ".  " + task_question
```

Object Motion Trajectory (x_{OMT})

```
"Consider following objects moving along (x, y, area) trajectories
in video to answer the question:" + \
", ".join(OMT_data) + ".  " + task_question
```

Table 5.8: **Prompt templates for three textual modalities.**

```
GOI_data = ["person", "oven", "dishwasher", "sink", "countertop",
"dish", "box", "scissors", "drain", "hand", "stove"]
```

```
OSL_data = ["stove located at (0.52, 0.64, 0.595)",
            "sink located at (0.56, 0.64, 0.211)",
            "countertop located at (0.63, 0.79, 0.308)",
            "box located at (0.46, 0.65, 0.142)",
            "dishwasher located at (0.5, 0.5, 0.991)",
            "dish located at (0.41, 0.75, 0.077)",
            "person located at (0.47, 0.76, 0.282)" ]
```

```
OMT_data = ["stove trajectory:  (0.5,0.5,0.991)->(0.51,0.69,0.397)
            ->(0.54,0.73,0.396)",
            dish trajectory:  (0.55,0.62,0.096)->(0.11,0.65,0.079)",
                                .
                                .
                                .
            "dish trajectory:  (0.41, 0.75, 0.077)",
            "person trajectory:  (0.54,0.81,0.34)->(0.49,0.72,0.339)
            ->(0.54,0.84,0.157)->(0.23,0.71,0.176)
            ->(0.51,0.79,0.232)->(0.52,0.78,0.266)
            ->(0.39,0.64,0.558)->(0.54,0.82,0.184)"]
```

Table 5.9: **Prompt examples for three textual modalities.**

Table 5.10: **Modality Constrained Variants on Robotics Domain:** We evaluate our modality constrained baselines on the robotics domain tasks and report accuracy (%). Note that a weighted sum over multiple tasks is reported here (similar to Table 5.4). Note the minimal increase over random for the variants in contrast to generic video benchmarks.

Method	Visual	Frames	Accuracy
Random	-	-	22.1
Just-LLM	✗	-	21.9
SF-VLM	✓	1	23.5
MVU	✓	16	30.4

```
prompt_list = \
    [f"Response {idx}:{val}" for idx, val in enumer-
ate(prompt_list)]

system_prompt = \
    "Considering given frames of a long video, select the most
     suitable response to the following question from the five
     options provided."

response_template = \
    "The correct response best answering the question about
the given
     video is "

task_prompt = "Question:  {qs}" + ".join(prompt_list)

qs = system_prompt + task_prompt + response_template
```

Table 5.11: **Likelihood Selection Sample Prompt Templates.** Variables *qs* and *prompt_list* refer to per sample question and choice list respectively.

Table 5.12: **Ablating Answer Candidates in Prompt:** We illustrate the importance of appropriate prompting when combining with likelihood selection, specifically for long video QnA tasks. Top-1 accuracy (%) is reported on EgoSchema subset (ES-S) and NextQA test set (NQA-T).

Dataset	ES-S	NQA-T
No answer candidates in prompt	58.2	35.8
With answer candidates in prompt	60.3	55.4

Table 5.13: **Open-Ended Video QnA Evaluation**: We present results on the ActivityNet dataset [303] that demonstrate strong performance of our proposed MVU framework. Accuracy (%) is reported. VT stands for video level training. We highlight how our MVU framework utilizes no video level training for any of its components and surpassed multiple approaches that rely on video-language training.

Method	Zero-Shot	VT	ActivityNet-QA
JustAsk [291]	✗	✓	38.9
FrozenBiLM [292]	✗	✓	43.2
VideoCoCa [289]	✗	✓	56.1
FrozenBiLM [292]	✓	✓	24.7
Video Chat [134]	✓	✓	26.5
LLaMA Adapter [319]	✓	✓	34.2
Video LLaMA [316]	✓	✓	12.4
Video-ChatGPT [160]	✓	✓	35.2
LocVLM [206]	✓	✓	37.4
Video-LLaVA [143]	✓	✓	37.4
VISTA-LLaMA [159]	✓	✓	37.4
VideoChat-2 [135]	✓	✓	37.4
LLaMa-VID [141]	✓	✓	37.4
LLoVi [314]	✓	✗	41.8
MVU (ours)	✓	✗	**42.2**

Table 5.14: **LongVideoBench Evaluation:** We integrate MVU with the baseline from [1] and highlight the additional performance improvements achieved by our MVU framework.

Method	Acc (%)
Phi-3-Vision-Instruct [1]	49.7
Phi-3-Vision-Instruct + MVU	50.4

Table 5.15: **Ablation on Object Motion Trajectory (OMT) modality:** We perform this ablation on a different dataset given the motion focused aspect we explore. Accuracy (%) reported on the motion-based SSv2 dataset clearly indicate the usefulness of the OMT modality in our MVU framework.

Method	OMT	Accuracy
Random	-	0.6
CLIP [196]	-	4.0
MAXI [145]	-	6.4
MVU (ours)	✗	3.6
MVU (ours)	✓	**7.2**

Table 5.16: **Frame Count Ablation:** We illustrate the importance of appropriate prompting when combining with likelihood selection, specifically for long video QnA tasks. Top-1 accuracy (%) is reported on EgoSchema subset (ES-S) and NextQA test set (NQA-T).

Method	Frames	EgoSchema-S	Time (s)
MVU	16	60.3	2.42
MVU	32	60.4	2.48
MVU	64	60.4	2.60
MVU	128	61.2	2.81

Table 5.17: **Context Length Comparison:** We compare the context length used (i.e. number of tokens) to achieve similar results with LLoVi [314] as opposed to our MVU. We achieve better performance utilizing less tokens.

Method	Average Tokens	ES-F (%)
LLoVI	1940	33.5
MVU	1124	37.6

Table 5.18: **Multi-Frame LLaVA Baseline:** We implemented multi-frame variants of LLAVA [149] with no video level training. Results indicate that without any video level training such naive extension does not lead to results improvements.

Method	Frames	ES-S
LLaVA	1	55.8
LLaVA	8	53.4
LLaVA	16	46.2
LLaVA	32	40.2

Chapter 6

Pixel Motion as Universal Representation for Robot Control

6.1 Introduction

Translating open-ended natural language instructions into robot actions is a cornerstone of flexible robot control. We identify two key requirements to enable this: (i) universal representations that support operating diverse embodiments [173, 213, 327], and (ii) benefiting from large-scale video-language data without action labels [23, 63, 80, 118]. We explore their intersection, proposing LangToMo, a vision–language–action framework structured as a *dual-system architecture*, inspired by dual-process theories of cognition [110] and recent hierarchical robotics frameworks [18, 24, 98, 177, 228]. In our high level *System 2* module, we use pixel motion as the robot action representation. We use image diffusion to learn to predict pixel motion from a single image (initial observation) conditioned on a language described action. Subsequently, our embodiment-specific low level *System 1* deterministically projects these action representations into executable robot actions.

We adopt pixel motion—the apparent motion of pixels between frames—as our *universal motion representation*, because it is agnostic to embodiments, viewpoints, and tasks. By predicting pixel motion instead of full RGB images, LangToMo captures essential motion patterns more efficiently than text-to-video generation [23, 63, 80, 118]. Pixel motion can be freely computed from videos using self-supervised methods like RAFT [244], enabling scalable, weakly supervised training on large video-caption datasets, similar to prior work on predictive world models [23, 80].

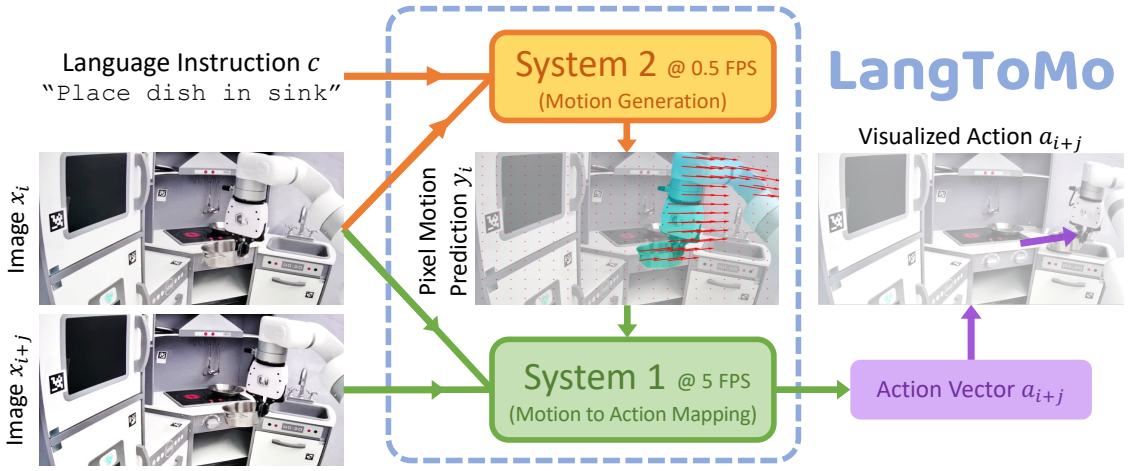

Figure 6.1: Illustration of proposed dual-system VLA framework, LangToMo, with pixel motion representations.

Optical flows, essentially a set of pixel motion (PM) between two consecutive frames, has been leveraged to enhance motion-focused video generation [119, 142]. Ko et al. [118] calculates flow from frame pairs to perform robot control, establishing the promise of this direction for robotics. In contrast, we directly generate PM from language and a single frame using our System-2 module, offering greater efficiency and performance. Our predicted PM serves as an interpretable intermediate representation for downstream systems (e.g., our System-1), enabling even unsupervised control via hand-crafted mappings. Alternate motion signals in image-space are used in works like [96, 227, 229, 238], but they rely on explicit dense annotations limiting training scalability, unlike our System-2 formulation.

Sequences of PM generated by our System 2 are then be transformed into robot actions via *System 1*, a fast and deterministic controller. Specifically, System 1 consists of task-specific action mappings tailored to different embodiments and viewpoints. We explore two instantiations of System 1: (a) learning mappings directly from limited expert demonstrations, and (b) hand-crafting mappings by leveraging the interpretable nature of pixel motion (motivated by [118]). Connecting System 1 and System 2 forms our overall language-conditioned robot control framework, LangToMo. This hierarchical formulation allows operating the expensive high-level System 2 at sparse temporal intervals while invoking the lightweight low-level System 1 at dense temporal intervals for efficient control.

In summary, our contributions are as follows:

- **Universal Action Representation:** pixel motion as a learnable, interpretable,

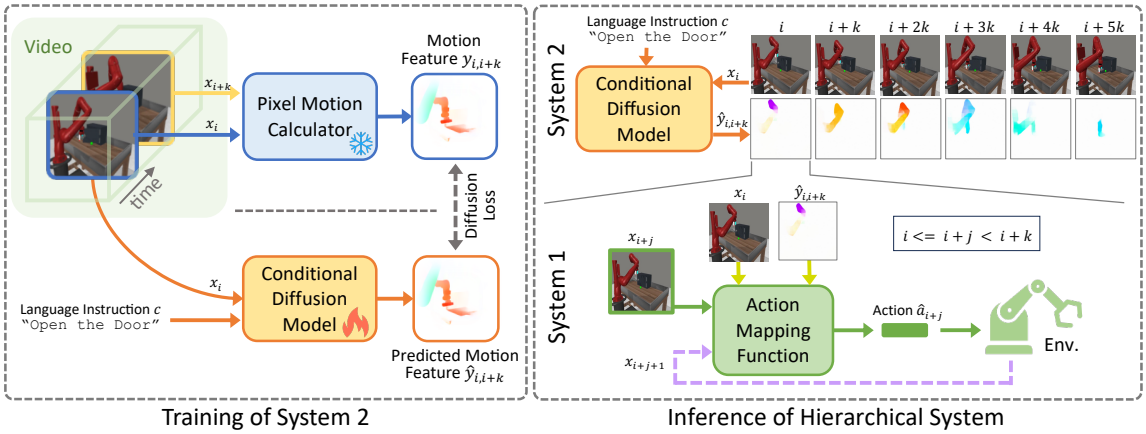

Figure 6.2: **Overview of LangToMo:** (Left) We learn to forecast pixel motion as universal motion features from video-caption pairs using scalable, self-supervised training of a diffusion model. (Right) Our *System 2* forecasts motion at sparse intervals (k), while *System 1* maps it to dense action vectors at j intervals ($j < k$).

and motion-focused representation for robot control tasks.

- **Simple & Scalable Learning:** mapping natural language actions to motion representations (pixel motion sequences) with a conditional diffusion model trained on web-scale video-caption data, without requiring pixel-level or action trajectory annotations.
- **Robotics Application:** conversion of learned action representations into action policies with minimal supervision, enabling operation under zero-shot and even unsupervised settings.

We evaluate LangToMo on both simulated and real-world environments, highlighting its effectiveness and generality across diverse robot control tasks.

6.2 Methodology

We tackle the problem of robot control from natural language instructions by introducing a two-stage framework. Language and visual inputs are first encoded into pixel motion based representations, which are then decoded into robot actions. This dual-system architecture comprises: *System 2*, a conditional image diffusion model that generates motion at sparse temporal intervals as a high-level controller; and *System 1*, a task-specific low-level controller that maps these pixel motions to executable action vectors. An overview of our framework, LangToMo, is shown in

Figure 6.2.

6.2.1 System 2: Pixel Motion Forecast

Optical flow estimation from frame pairs is a well-defined problem (exact solutions exist) that has been extensively studied [152, 156, 244, 282]. In contrast, estimating pixel motion (PM) from a single image and language instruction is inherently multi-modal: a caption-frame pair may correspond to multiple valid flows, each representing a different trajectory toward the goal. We use this challenging task as our self-supervision objective: learning a mapping from *language to motion*. Furthermore, we incorporate temporal context by conditioning on the motion of a previous state.

Consider a video clip $\boldsymbol{x} \in \mathbb{R}^{t \times h \times w \times c}$ with t, h, w, c for frames, height, width, and channels respectively. Also consider an embedding vector, $\boldsymbol{c}$ representing the paired caption for that clip. Denoting the i-th frame of video as $\boldsymbol{x}_i$, we define pixel motion, $\boldsymbol{y}_{i,i+k}$, that corresponds to motion between frames $\boldsymbol{x}_i \rightarrow \boldsymbol{x}_{i+k}$ where k is a constant. Our language to motion mapping function, $\mathcal{D}$ becomes,

$$\hat{\boldsymbol{y}}_{i,i+k} = \mathcal{D}(\boldsymbol{x}_i, \boldsymbol{y}_{i-k,i}, \boldsymbol{c} \mid \theta) \tag{6.1}$$

where $\hat{\boldsymbol{y}}_{i,i+k}$ is the predicted motion representation from the i-th state to $(i+k)$-th state *without* knowing $\boldsymbol{x}_{i+k}$. θ are learnable parameters.

We reiterate the multi-modal output aspect of our mapping described in Equation (6.1) (i.e. one to many mapping due to multiple optimal $\hat{\boldsymbol{y}}_{i,i+k}$). Diffusion models have shown excellent abilities to model such distributions [44, 56]. Considering the 2D structure present in our images and pixel motion, for $\mathcal{D}$ we elect to utilize a 2D conditional U-Net based diffusion model [199] operating at pixel level. Our goal is to learn a set of parameters, θ for this diffusion model based mapping as,

$$\arg\min_{\theta} \|\boldsymbol{y}_{i,i+k} - \mathcal{D}(\boldsymbol{x}_i, \boldsymbol{y}_{i-k,i}, \boldsymbol{c} \mid \theta)\|_2 \tag{6.2}$$

that allows our language to motion mapping to perform instruction based robot control. Next we dive into the learning process of our diffusion based implementation for this mapping function.

6.2.2 Diffusion based Motion Representation Learning

Background: Diffusion Models generate data by progressively denoising corrupted signals, optionally conditioned on a goal input. While inference follows this

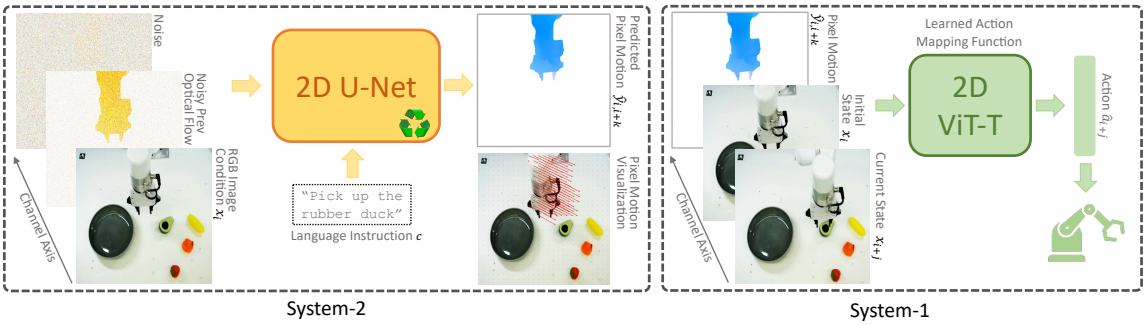

Figure 6.3: **LangToMo Architecture:** (Left) Diffusion model generates pixel motion conditioned on RGB image, prior motion, and caption. Visualized predictions are overlaid as arrows. (Right) ViT-T network maps predicted motion to robot actions in supervised setting, conditioned on initial/current states and target motion.

iterative refinement process, training is conducted more efficiently using parallel denoising steps: the model is trained to predict less noisy versions of intermediate corrupted signals generated from clean data, a procedure analogous to teacher forcing (more details in Section 6.5.4).

Architecture: The defacto architecture for diffusion based conditional image generation is the 2D conditional U-Net [216], which maps between 2D RGB images with an embedding based conditioning through cross-attention in the model intermediate layers. Basing off this setup, we modify the input and output heads to process 7 and 2 channel tensors respectively (instead of default 3 channel RGB). Two of the input channels and the two output channels correspond to our pixel motion target (noise input and clean output). The remaining 5 input channels correspond to our 2D-structured conditions: previous pixel motion (2 channels) and current state image (3 channels). These conditional inputs are not subject to the standard noise corruption schedule during training or inference (details in Section 6.5.4). The textual embedding is provided as the default embedding condition. Our channel modification to accommodate additional structured conditions allows a minimal design, retaining the general structure of the U-Net that is known to excel at 2D generative modeling. Such input channel concatenation based conditioning has been used in diffusion literature for different tasks [89, 220] and is inspiration for our design. We illustrate this architecture in Figure 6.3 (left).

Calculating Pixel Motion Ground-truth: We utilize the RAFT algorithm [244] to calculate our target pixel motion $\boldsymbol{y}_{i,i+k}$, using frames $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+k}$. This is an efficient iterative algorithm that calculates a good estimate of optical flow, in other words, pixel motion. Each pixel motion, $\boldsymbol{y}_{i,i+k} \in \mathbb{R}^{h \times w \times 2}$, contains two channels for spatial

directions, that are normalized to a $(0, 1)$ range. All motion is represented within this 2D space - extensions to a third depth dimension are left as a future direction. Our experiments indicate the sufficiency of such 2D spaces to encode motions relevant to robot actions. We note that given the presence of background motions in both natural and simulation images (e.g. shadows moving with objects), this target pixel motion contains noise that is not directly relevant to the underlying motion, underscoring the challenging nature of our self-supervision objective.

Previous Pixel Motion Representation: The other input signal to our mapping function is past frames pixel motion. Motivated by success of teacher forcing in generative modeling of both language [195] and videos [233], we use the target pixel motion of previous time steps during our System-2 training. We also note the importance of representing pixel motion relative to current state as our mapping function is conditioned on the current image (details in Section 6.5.2). Similar findings are observed in image-pair based optical flow calculation literature [220].

Language Instruction Embeddding: The primary input conditioning of our mapping function is the natural language based action description that is used to control the generated motions. Following prior robotics literature [182], we use a Universal Sentence Encoder model [34] to convert textual instructions to fixed size embedding vectors. This embedding model is trained to capture sentence level meanings. We use an off-the-shelf pretrained version, keeping all model parameters unchanged (more details in Section 6.5.3).

Training: Our training uses the standard diffusion denoising objective [88] between predicted ($\hat{\boldsymbol{y}}_{i,i+k}$) and target ($\boldsymbol{y}_{i,i+k}$) pixel motion. The conditional 2D inputs, $\boldsymbol{x}_i$ and $\boldsymbol{y}_{i-k,i}$ are not subject to a noising schedule. The image condition, $\boldsymbol{x}_i$, remain uncorrupted while the previous pixel motion, $\boldsymbol{y}_{i-1,i}$, is set to random noise or a partially corrupted version to align with inference settings. We also introduce zero motion to ends of videos such that when textual instruction is complete, those visual states map to zero motion. More details in Section 6.5.4.

Inference: We forecast pixel motion from i to $i + k$ timestamp using a 25-step DDIM schedule with only the current image observation $\boldsymbol{x}_i$. At the initial step, the model only takes the image $\boldsymbol{x}_i$ (state observation), language instruction c, and random noise as the previous pixel motion. For subsequent steps, the previously predicted motion is reused, enabling sequential pixel motion generation that drives the system toward fulfilling the language command.

6.2.3 System 1: Pixel Motion to Action Mapping

Our System 2 produces pixel motion conditioned on a given state-instruction pair. We next detail how these pixel motion representations are mapped into action vectors that directly control the robot. Consider a mapping function, $\mathcal{F}$, operating at dense temporal intervals:

$$\hat{\boldsymbol{a}}_{i+j} = \mathcal{F}\left(\hat{\boldsymbol{y}}_{i,i+k}, \boldsymbol{x}_i, \boldsymbol{x}_{i+j}\right), \tag{6.3}$$

where $j \in [0, k]$, i is a multiple of k (for a hyperparameter k), and $\hat{\boldsymbol{a}}_{i+j}$ denotes the predicted action vector for the $(i+j)$-th state. An overview of this formulation is shown in Figure 6.2 (right).

While *System 2* is trained as a general-purpose motion generator across diverse embodiments, viewpoints, and environments, action vectors $\boldsymbol{a}_i$ are inherently embodiment-specific. Hence, we design *task-specific* mapping functions to serve as *System 1 (Action Mapping)*, converting pixel motion into executable robot actions.
Learned Mapping: We implement a neural network-based mapping function that can be trained using ground-truth action trajectories. Given the 2D spatial structure of the inputs to $\mathcal{F}$ (i.e., $\hat{\boldsymbol{y}}_{i,i+j}$, $\boldsymbol{x}_i$, $\boldsymbol{x}_{i+j}$), we channel-concatenate them and feed the resulting tensor to a lightweight vision transformer to predict action vectors. This architecture is illustrated in Figure 6.3 (right). The network is trained on a limited amount of task-specific demonstration data. Connecting this learned *System 1* with *System 2* following Equation (6.3), we obtain a complete pipeline for language-conditioned robot control. We refer to the resulting system, which uses a supervised learned mapping, as LTM-S.
Hand-Crafted Mapping: The interpretable nature of pixel motion also enables hand-crafted designs for $\mathcal{F}$. We refer to the resulting pipeline based on hand-crafted mappings as LTM-H. For simulated environments where ground-truth segmentations and depth maps are available, we follow the methodology in [118] to define action mappings, ensuring a fair evaluation of the utility of our pixel motion predictions compared to prior works. For real-world robot control, we construct viewpoint-specific hand-crafted mappings following [139]. Further details on both learned and hand-crafted mappings are provided in Section 6.5.5.

We highlight how our System 1 operates at a frequency different to our System 2, allowing a balance between efficiency and dense control. Our System 1 is also designed to be lightweight, given how it performs an almost deterministic mapping.

6.3 Experimental Results

We conduct experiments on 15 tasks spanning both simulated and real-world environments to highlight the strong performance of our proposed LangToMo framework. We also present multiple ablations to justify key design choices within our method.

Implementation Details: Our framework consists of *System 2 (Motion Generation)* containing a diffusion model, and *System 1 (Action Mapping)* containing either a learned or hand-crafted mapping function. We pretrain the diffusion model on a subset of the OpenX dataset [182], followed by optional fine-tuning on downstream task datasets. Pretraining is performed for 300,000 iterations with a learning rate of 1e-4, following a cosine learning rate schedule with 500 warmup steps, using 8 A100 GPUs (48GB) with a per-device batch size of 32 samples. Fine-tuning is performed for 100,000 iterations on 4 A5000 GPUs (24GB) with a batch size of 32 and a learning rate of 1e-5, again following a cosine schedule with 500 warmup steps. The learned action mapping (System 1) is trained separately using a vision transformer for 10,000 iterations on a single A5000 GPU with a batch size of 128 and a learning rate of 1e-4. During inference of our System 2 diffusion model, we use a DDIM scheduler with 25 steps to generate flow sequences, starting from noise. For each invocation of System 2, we run System 1 for 10 control steps (or until convergence in the hand-crafted setting). This hierarchical procedure is repeated until the episode terminates.

6.3.1 MetaWorld Simulated Environment

MetaWorld [302] is a simulated benchmark containing several robot manipulation tasks with accompanying natural language instructions. Each task episode corresponds to successfully completing an action described in language. The environment utilizes a Sawyer robot arm.

Training: We train *System 2* (diffusion model) first on the OpenX subset, followed by additional training on 165 MetaWorld videos (identical to the split used in [118]). For the learned variant of *System 1*, we train on 20 expert demonstrations per task. We also implement a hand-crafted variant of System 1, following the design in [118] to ensure fair comparison.

Evaluation: Following evaluation settings identical to [118], we evaluate each policy across 11 tasks. For each task, videos are rendered from 3 distinct camera poses, with 25 randomized trials (different initial positions of the robot arm and objects) for each view. We replicate multiple baselines from [63, 118] under

	door-open	*door-close*	*basketball*	*shelf-place*	*btn-press*	*btn-top*	*faucet-close*	*faucet-open*	*handle-press*	*hammer*	*assembly*	*Overall*
BC-Scratch	21.3	36.0	0.0	0.0	34.7	12.0	18.7	17.3	37.3	0.0	1.3	16.2
BC-R3M	1.3	58.7	0.0	0.0	36.0	4.0	18.7	22.7	28.0	0.0	0.0	15.4
UniPi (With Replan)	0.0	36.0	0.0	0.0	6.7	0.0	4.0	9.3	13.3	4.0	0.0	6.1
AVDC (Flow)	0.0	0.0	0.0	0.0	1.3	40.0	42.7	0.0	66.7	0.0	0.0	13.7
AVDC (Default)	72.0	89.3	37.3	**18.7**	60.0	24.0	53.3	24.0	81.3	**8.0**	6.7	43.1
LTM-H (Ours)	76.0	94.7	38.0	15.2	**82.0**	**84.7**	41.3	33.3	97.3	4.2	**6.9**	52.1
LTM-S (Ours)	**77.3**	**95.0**	**39.0**	**18.7**	**82.0**	84.3	**46.7**	**35.3**	**98.0**	6.7	**6.9**	**53.6**

Table 6.1: **Results on MetaWorld Environment:** We report the mean success rate across tasks. Each entry of the table shows the average success rate aggregated from 3 camera poses with 25 seeds for each camera pose.

common settings for comparison.
Results: We present the success rates for the 11 tasks and the average across tasks in Table 6.1. Both our LTM-H and LTM-S variants achieve strong overall performance, highlighting the effectiveness of our framework. Notably, several strong approaches [63, 118] exhibit moderate success rates, underscoring the difficulty of the benchmark. An important point of comparison is the AVDC (flow) baseline from [118], which also uses pixel motion prediction but differs in model architecture, flow representation, and training procedures. The improved performance of LangToMo over AVDC demonstrates the impact of our design choices.

6.3.2 Real-World Environment

We next evaluate on 4 challenging tasks in an xArm Table Top environment, constructed following the real-world setup in [139]. Examples of these tasks are shown in Figure 6.4. The tasks involve tabletop manipulations specified by language commands (details in Section 6.5.6).
Training: We train *System 2* (diffusion model) on the OpenX subset, followed by optional fine-tuning on 10 videos per task collected in the same real-world environment. We replicate the AVDC baseline [118] by training under identical conditions. All other baselines are implemented following the settings used in [139]. For *System 1*, we construct a hand-crafted mapping function based on [118, 139] (details

Method	Video Only Training	T1	T2	T3	T4	Avg
RT-2 Style [27]	✗	0	0	0	0	0
LLaRA [139]	✗	70	80	55	55	65.0
AVDC [118]	✓	10	20	0	0	15.0
LTM-H (ours)	✓	**80**	**70**	**65**	**60**	**68.8**

Table 6.2: **Real World Task Performance:** We follow the setup in LLaRA [139] to evaluate model performance on real world tasks under fine-tuned settings.

Method	Video Only Training	T1	T2	T3	T4	Avg
RT-2 Style [27]	✗	0	0	0	0	0
LLaRA [139]	✗	40	20	10	20	22.5
AVDC [118]	✓	0	0	0	0	0
GPT-4o [180]	✓	20	30	10	15	18.8
LTM-H (ours)	✓	40	30	35	30	33.8

Table 6.3: **Zero-Shot Transfer on Real World Tasks:** Evaluations follow settings in [139].

Table 6.4: **Ablation Study:** We report mean success rate (overall) on MetaWorld benchmark with our LTM-S variant. (left) Results highlight importance of key components in our System-2 model. (right) Results justify several high-level design choices of our framework.

Img	Lang	Prev Flow	PT	Overal	Method	Overall
✓	✓	✓	✓	53.6	Ours (default)	53.6
✓	✓	✓	✗	53.1	No diffusion	16.2
✓	✓	✗	✗	50.2	CA instead of concat	15.8
✓	✗	✗	✗	39.7	Sys-1 & 2 same freq	48.7
✗	✓	✗	✗	15.4	Only learned Sys-1	15.8

in Section 6.5.6).

Evaluation: We follow evaluation settings identical to [139], evaluating each policy across 4 tasks with a fixed camera view and 20 randomized trials per task. Each trial uses different initial positions of the objects present in the environment.

Results: We present results in Tables 6.2 and 6.3 to highlight the strong performance of LangToMo (baseline details in Section 6.5.7). The difficulty of these tasks is apparent by the moderate results from recent methods like LLaRA [139]. Notably, despite relying on heuristic-based hand-crafted mappings in *System 1*, LangToMo outperforms several state-of-the-art baselines such as RT-2 [27] and LLaRA [139], all without requiring action trajectory labels during training. Our framework learns directly from videos paired with natural language captions, showing the promise of this direction.

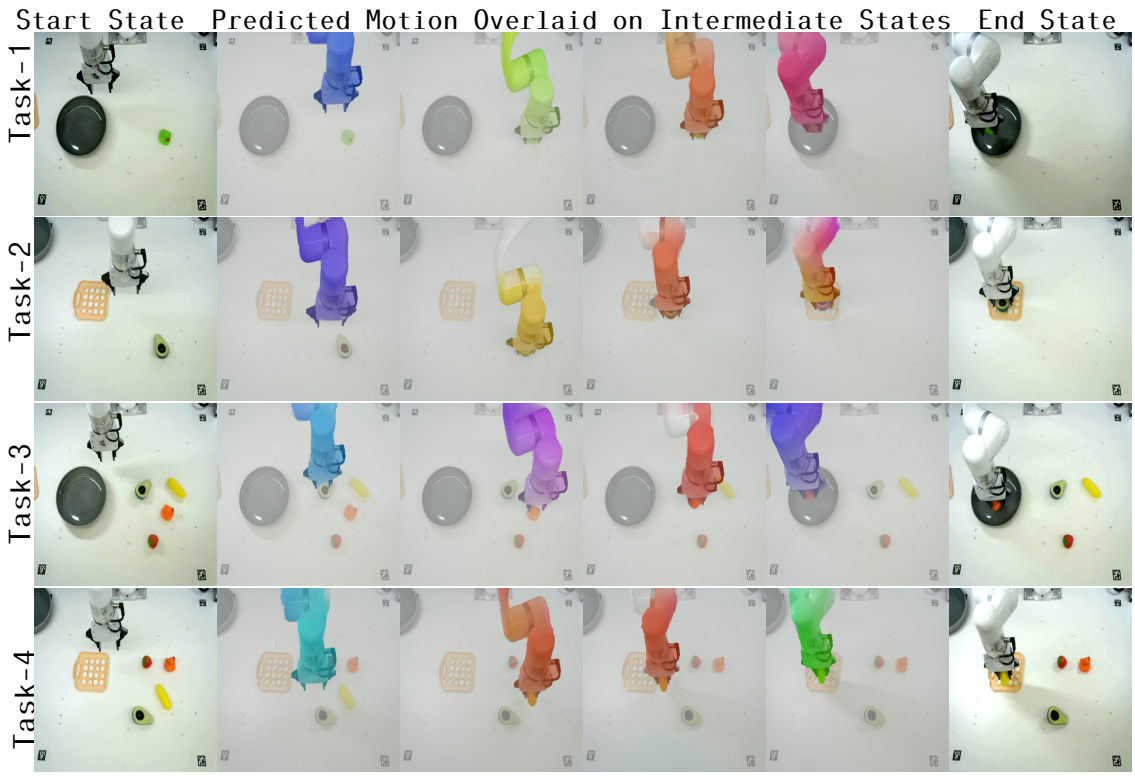

Figure 6.4: **Real World Tasks:** We illustrate the four real-world tasks following LLaRA [139]. Start and end states are shown in the first and last columns, with predicted pixel motion (color indicates motion direction) overlaid on intermediate states. LangToMo performs these challenging tasks successfully (see results in Table 6.2).

6.3.3 Ablation Studies

We conduct a series of ablative studies with LTM-S on the MetaWorld benchmark to evaluate the importance of key components within LangToMo. Results are summarized in Table 6.4.

System 2 Input Conditioning & Pretraining: Removing visual ("Img"), language ("Lang"), or previous flow ("Prev Flow") conditional inputs to the diffusion model significantly reduces performance, highlighting importance of each conditioning signal. On the other hand, removing diffusion model pretraining ("PT") leads to a modest performance drop, indicating that while pretraining aids convergence and performance, the framework remains effective with limited finetuning alone.

Simpler Baselines: Replacing diffusion ("No diffusion") with an autoencoder

breaks System-2 learning process. Modifying conditioning strategy to cross-attention ("CA instead of concat") also degrades performance. Skipping the iterative System-1 design (running System-1 at same frequency), and generating multiple actions per System-2 generated motion at once ("Sys-1 & 2 same freq") also degrades success rates, validating our design choices. Additionally, bypassing intermediate motion representations ("Only learned Sys-1") leads to poor results, underscoring the necessity of our two-stage architecture. See Section 6.5.8 for a detailed discussion.

6.4 Conclusion

We presented LangToMo, a scalable vision-language-action framework that decouples motion generation and action execution through a dual-system architecture. By leveraging diffusion models to learn universal pixel motion representations from video-caption data, our *System 2* enables generalizable, interpretable motion planning without dense supervision. These motions are translated into robot actions by *System 1*, using either learned or hand-crafted mappings tailored to specific embodiments. Extensive experiments across simulated and real-world environments demonstrate strong performance of LangToMo, highlighting the promise of universal motion representations as a bridge between language, vision, and action for scalable robot learning.

Limitations

LangToMo is pretrained on large-scale video-caption data, but relies on hand-crafted or learned action mappings in System 1 which can be costly for each new downstream task. Learning robust, transferable mappings remains an open challenge. Also, our framework models motion using 2D pixel motions, which currently lacks depth cues. Extending to 3D motion representations is left as a future direction. In terms of speed, despite operating at sparse intervals, System 2 relies on diffusion models that remain computationally expensive at inference time, limiting use in resource-constrained deployments. This is another future direction we hope to explore further. Finally, we currently do not account for ego motion in training videos: we limit our training to fixed camera videos (no ego motion). A key next direction is extending our System-2 training to include videos with ego motion, which would allow scaling to any kind of video.

6.5 Additional Details

6.5.1 Additional Experimental Results

We first present additional results on our real world environment, focused on highlighting the usefulness of human demonstrations for our method. A key benefit of our pixel motion based control (similar to prior work AVDC [118]) is the ability to learn from human demonstrations directly (with no requirement for keypoint based remapping or other dense annotations). We investigate this aspect of our proposed LangToMo first, presenting results in Table 6.5. Results indicate clear usefulness of incorporating human demonstrations in addition to robot demonstrations, as well as the ability to learn from human demonstrations directly. We illustrate some examples of human and robot demonstrations used for training in Figure 6.5.

Method	Data	T1	T2	T3	T4	Average
AVDC	RD	10	20	0	0	15.0
LTM-H (ours)	RD	80	70	65	60	68.8
LTM-H (ours)	HD	40	35	40	30	36.3
LTM-H (ours)	RD+HD	80	75	65	65	71.3

Table 6.5: **Extended Results on Real World Environment:** We evaluate the impact of using human demonstrations (HD) in addition to robot demonstrations (RD) as training data for our System 2 diffusion model. The standard setting following prior work is training on RD. AVDC trained on RD is provided as a baseline. Results indicate that training our method on HD alone performs reasonably, while using HD along with RD fortraining boosts performance further. RD here refers to human controlled robot demonstrations while HD refers to human controlled human demonstrations (e.g. human using their own human hand to move an object) See Figure 6.5 for examples.

We next explore the ability to extend our method to benchmarks that involve ego motion of the robot (e.g. simple navigation tasks). Following prior work AVDC [118], we evaluate on the iThor benchmark and present results in Table 6.6. Results indite clear improvements of our proposed LangToMo over naive baselines and prior work AVDC [118].

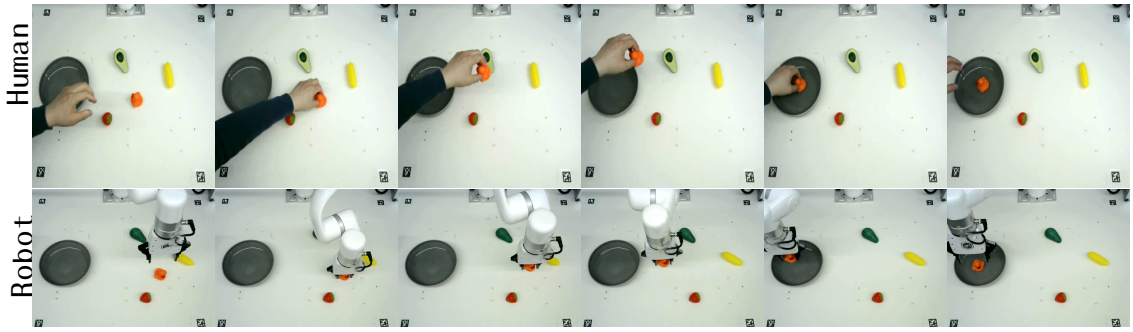

Figure 6.5: **Human and Robotic Demonstrations:** We visualize frames from videos of two sample demonstrations on our real world environment. These human (top) and robot (bottom) demonstrations can both be used to fine-tune our System-2 diffusion model, highlighting a unique aspect of our hierarchical LangToMo approach. Both examples use the common caption of "`Pick up the rubber duck and place on the bowl.`"

6.5.2 Relative Pixel Motion

A key design choice in our formulation is to represent pixel motion with respect to the current frame ($\boldsymbol{x}_t$), rather than the previous frame ($\boldsymbol{x}_{t+1}$) or some other frame. This aligns with the structure of our conditional diffusion model, which receives $\boldsymbol{x}_t$ as a secondary conditioning input. Predicting the transformation from $\boldsymbol{x}_t$ to the next frame allows the model to more directly focus on the visual cues present in the current state. In contrast, predicting motion from $\boldsymbol{x}_{t-1}$ or some other different frame would require indirect reasoning over a non-visible state, introducing additional complexity. Hence our approach is to represent past pixel motion (e.g. $\boldsymbol{x}_{t-1}$ to $\boldsymbol{x}_t$) as $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t-1}$ instead. While this may seem counterintuitive, we note how prior literature on image-pair-based optical flow prediction for video tasks has also found that defining motion in terms of a reference image—particularly the current frame that is visible—can lead to more stable and accurate flow estimates [142]. Moreover, our experiments representing previous motion in a different manner lead to subpar performance, standing as further evidence.

We also experiment trying to predict an additional future motion relative to a future frame. We compare this against predicting that same future motion relative to the current frames. In this setting, the latter performs well while the former variant fails to learn meaningful motion signals predictions.

Method	Kitchen	Living Room	Bedroom	Bathroom	Overall
BC-Scratch	1.7	3.3	1.7	1.7	2.1
BC-R3M	0.0	0.0	1.7	0.0	0.4
AVDC	26.7	23.3	38.3	36.7	31.3
LTM-H (ours)	27.3	23.7	40.0	36.7	31.9

Table 6.6: **Results on iThor Benchmark:** We follow the iThor dataset based evalution setup used in AVDC paper to demonstrate that our method generalizes to robot movement based control as well (i.e. where ego motion occurs). Results indicate that our method outperforms AVDC across categories and overall.

6.5.3 Language Embedding Model

For the language embedding model, we employ the Universal Sentence Encoder (USE), a pre-trained model from [34]. USE generates fixed-length vector representations of text, capturing rich semantic meaning, making it suitable for various natural language processing (NLP) tasks. Its widespread use in research, including works like OpenX [182], highlights its effectiveness in transforming textual input into meaningful embeddings even for robotic tasks. In our framework, the USE serves as a key component, encoding language instructions into dense vectors that are later used to guide the generation of motion representations. The model's ability to produce consistent and high-quality embeddings enables seamless integration between language and vision modalities, ensuring that our system can accurately interpret and respond to diverse language commands.

6.5.4 Diffusion Model Details

In our diffusion model training, input noising is applied by adding Gaussian noise to the target motion data (following standard settings [88]). The image condition input and the previous flow are not subject to this noising. The previous flow is corruption with a 50% chance. During corruption, a random amount of Gaussian noise is added. To ensure diverse and meaningful training, filtering and augmentation operations are performed on the frames as described next. The indices corresponding to consecutive frames (i and $i+1$) are selected such that they maintain fixed intervals based on the video frame rate. Frames with zero optical flow (i.e., no motion) between i and $i+1$ are filtered out to avoid irrelevant data. Additionally, to handle the completion of textual instructions, we introduce zero

motion at the ends of videos, ensuring that these states map to a lack of motion when the instruction concludes. The visual inputs (images and optical flow) are cropped and resized, with appropriate transformations applied to the flow data to maintain consistency.

6.5.5 Hand-Crafted Mapping Functions

Synthetic Environments: We follow the formulation of [118] using a segmentation map of robot controller and a depth map of environment. The generated pixel motions are converted into directions in 3D space to move the robot controller based on these dense maps. We direct the reader to Ko et al. [118] for further details.
Real World Environments: Following Li et al. [139], we build our real world environment with a single plane assumption (e.g. table top manipulation) and map the predicted pixel motions for the robot controller center points onto the plane (using visual geometry). An initial camera calibration is performed for the environment to obtain necessary camera matrices. After extracting a start and end position for a manipulation task following this setting, our position to action vector conversion is identical to [139].

6.5.6 Real World Experiments

We perform four types of real world experiments as illustrated in Figure 6.4. The language instructions for the four tasks are as follows:

1. `Pick up the duck and place on the bowl.`
2. `Pick up the duck and place on the tray.`
3. `Pick up the avocado and place on the bowl.`
4. `Pick up the corn and place on the tray.`

We select these following Li et al. [139] to ensure fair comparisons to prior works.

6.5.7 Baseline Details

Our key baselines are from AVDC [118] and LLaRA [139]. For both methods, we use their official implementations to replicate their results and evaluate ours under identical settings. For LLaRA, all results are reported on their inBC variant for fair

comparison against our method (i.e. similar inputs during inference / no external scene object information).

6.5.8 Detailed Ablations

We discuss our ablations in Table 6.4 in detail in the following section.
System 2 Design Choices: We first ablate critical inputs to *System 2 (Motion Generation)*. Removing pretraining ("PT") leads to a modest performance drop (from 53.6% to 53.1%), indicating that while pretraining aids convergence, the framework remains effective with limited finetuning alone. Removing the previous optical flow input ("Prev Flow") results in a larger decline to 50.2%, validating the importance of temporal conditioning. Ablating the language embedding leads to a significant drop (to 39.7%), highlighting the necessity of semantic instruction guidance. Finally, removing the visual input ("Img") results in near-random performance (15.4%), confirming that visual grounding is essential.
High-Level Framework Design: We next evaluate several higher-level architectural decisions. Removing the diffusion model ("No diffusion") and training a direct regressor leads to a sharp performance drop (to 16.2%), underscoring the value of iterative, probabilistic modeling for motion generation. Replacing input concatenation with cross-attention ("CA instead of concat") similarly degrades performance, suggesting that simple spatial concatenation is a more effective conditioning strategy for our setting. Using a multi-action decoder within *System 1* to run it at same frequency as our system 2 ("Sys-1 & 2 same freq") results in slightly lower performance (48.7%), indicating that our default action mapping is more effective. Training only a learned *System 1* without leveraging pre-generated optical flows ("Only learned Sys-1") performs poorly (15.8%), demonstrating that direct action generation without intermediate motion representation is insufficient for generalization.

Chapter 7

Conclusion and Future Work

In this thesis, we explored how language can be leveraged to guide and improve video representation learning. Inspired by how humans build internal models of the world through multi-sensory inputs, we proposed a framework that connects natural language supervision to video understanding with minimal supervision from task specific human annotations. Our approach unfolds over four stages, each addressing a distinct gap in the video understanding pipeline.

We began by introducing Language-based Self-Supervised Learning (LSS) for videos, which adapts CLIP-style image-language models to the video domain through the construction of language-aligned concept spaces. LSS learns from unlabelled video data while retaining the generality and zero-shot capabilities of image-language pretraining.

In LocVLM, we addressed the lack of spatial awareness in multimodal language models. By representing object locations as natural language coordinates and integrating them through instruction tuning, we significantly improved spatial reasoning, reduced object hallucination, and enabled precise region-level localization and description. Our video QnA evaluations highlight how such improved spatial awareness leads to clear improvements in complex video understanding tasks.

Extending our scope further to tackle the challenges of reasoning over long videos, we proposed Multimodal Video Understanding (MVU) framework. This framework extracts object-centric information—global presence, spatial location, and motion trajectories—and expresses them in natural language. We define this as our language based representation for object motions. These motion representations allow off-the-shelf LLMs to perform long-range reasoning about actions and interactions, improving performance on temporally complex video tasks.

Finally, we introduced LangToMo, an approach to modeling motion that bridges

natural language and vision with structured motion representations. By learning to map language to pixel-level motion using internet-scale video-caption data, LangToMo enables temporally grounded understanding. Going beyond evalutions on visual datasets, we explored real world interaction based downstream tasks such as robot control.

Together, these contributions form a unified language-grounded video learning framework that is self-supervised, scalable, and generalizable across diverse video understanding problems. Our experiments demonstrate improvements in video classification, question answering, spatial localization, and robotic interaction—highlighting the value of integrating language priors with structured video representations.

7.1 Future Work

Looking ahead, we identify several directions for extending this work. First, we aim to move beyond 2D pixel motion and toward *explicit modeling of motion in 3D*. Incorporating depth and geometric reasoning can support richer understanding of actions, object dynamics, and physical interactions. This opens the door to more generalizable and physically grounded models, especially important for tasks such as embodied AI and robotics.

Second, we propose to expand our training and evaluation to *videos with ego motion*, such as first-person or mobile-camera footage. Handling ego motion introduces challenges in disentangling camera and object movement, but it is critical for building robust, real-world video understanding systems. Developing motion-invariant representations or explicitly modeling egocentric geometry will be key in this direction.

Together, these directions will continue the trajectory of this thesis: toward language-aligned, structured, and grounded models of the visual world that approach the abstraction and reasoning capabilities of humans.

7.1.1 3D Aware Motion Modeling

One promising direction for extending this work is the explicit modeling of motion in 3D. While our current approach focuses on 2D pixel motion trajectories, real-world actions and interactions unfold in three-dimensional space. Understanding such motion in 3D is essential for more generalizable and physically grounded representations—particularly in domains like embodied AI, robotics, and augmented reality.

To this end, we propose a pipeline that begins by leveraging single-image or video-based *depth estimation* methods to project 2D pixel motion trajectories into 3D space. This process will transform observed pixel displacements into camera-aligned 3D trajectories, taking into account geometric structure and depth cues.

We will then design a *structured representation* of these 3D motion trajectories that captures their spatio-temporal dynamics in a compact and learnable format. This representation will serve as the output domain for a generative model trained to predict motion given visual and language context.

To model the distribution of plausible 3D motion trajectories, we plan to use a *diffusion model*, which has shown strong performance in structured generative tasks. By conditioning the diffusion process on both visual state and language input, we aim to generate coherent and grounded 3D motions that reflect both observed scene structure and high-level intent.

This direction not only enhances the expressiveness of our framework but also strengthens its applicability in real-world scenarios where depth, physicality, and interaction matter. It represents a key step toward building systems that perceive, reason about, and act in the world with the richness and flexibility of human understanding.

7.1.2 Compatibility with Ego Motion in Videos

Another important direction is the expansion of our framework to handle *ego motion*—the motion of the camera itself—as commonly encountered in first-person videos, drone footage, or mobile robotic platforms. Unlike static-camera settings, videos with ego motion present a significant challenge: distinguishing between object motion and camera-induced motion becomes non-trivial. Yet, this setting is essential for robust real-world video understanding and action grounding. This is also a task simple for humans, that in fact happens almost unconsciously.

We propose to address this by estimating camera motion using established ego-motion estimation or structure-from-motion algorithms. Given any video, we aim to compute the camera's trajectory and subsequently represent pixel or object motion *relative to the camera frame*. This enables the disentangling of observed dynamics into camera-centric motion representations, allowing the model to focus on true object or agent behavior.

In one setup, ego motion compensation will be applied only during training. That is, motion will be normalized with respect to camera movement during training to encourage invariance, while inference will be conducted on videos with

fixed cameras. This training regime allows the model to benefit from any video sources without limitations to fixed camera assumptions in training videos.

We also plan to explore a fully generalizable setup where ego motion is present during both training and inference. In this case, the model must learn to reason about motion *relative to the current camera state*, effectively building an internal frame of reference. This direction may involve conditioning motion prediction on estimated egocentric pose or explicitly modeling camera-centric coordinate systems.

By developing motion-invariant representations and egocentric modeling capabilities, our framework can extend to more diverse and unconstrained video domains, enhancing generalization to real-world deployment scenarios in robotics, augmented reality, and embodied learning.

Bibliography

[1] M. Abdin et al. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv*, abs/2404.14219, 2024. URL https://api.semanticscholar.org/CorpusID:269293048. 74, 83

[2] J. K. Aggarwal and M. S. Ryoo. Human activity analysis. *ACM Computing Surveys (CSUR)*, 43:1 – 43, 2011. URL https://api.semanticscholar.org/CorpusID:5388357. 11, 40

[3] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *ICCV*, 2015. 5

[4] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *ArXiv*, abs/2104.11178, 2021. 21

[5] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a Visual Language Model for Few-Shot Learning. *NeurIPS*, 2022. 6, 7, 41

[6] B. AlKhamissi, M. ElNokrashy, M. AlKhamissi, and M. Diab. Investigating cultural alignment of large language models, 2024. 8

[7] M. Argus, L. Hermann, J. Long, and T. Brox. Flowcontrol: Optical flow based visual servoing. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7534–7541, 2020. URL https://api.semanticscholar.org/CorpusID:220280145. 9

[8] G. Arnold. Rheotropism in fishes. *Biological Reviews*, 49, 1974. URL https://api.semanticscholar.org/CorpusID:30755969. 9

[9] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, J. Jitsev, S. Kornblith, P. W. Koh, G. Ilharco, M. Wortsman,

and L. Schmidt. Openflamingo, Mar. 2023. URL https://doi.org/10.5281/zenodo.7733589. 6, 7

[10] S. Bahl, A. Gupta, and D. Pathak. Human-to-robot imitation in the wild. In *Robotics: Science and Systems*, 2022. 9

[11] M. Bain, A. Nagrani, G. Varol, and A. Zisserman. A CLIP-Hitchhiker's Guide to Long Video Retrieval. *arXiv preprint arXiv:2205.08508*, 2022. 5

[12] E. Baird, N. Boeddeker, and M. V. Srinivasan. The effect of optic flow cues on honeybee flight control in wind. *Proceedings of the Royal Society B*, 288, 2021. URL https://api.semanticscholar.org/CorpusID:231643236. 9

[13] I. Balavzevi'c, Y. Shi, P. Papalampidi, R. Chaabouni, S. Koppula, and O. J. H'enaff. Memory consolidation enables long-context video understanding. *ArXiv*, abs/2402.05861, 2024. URL https://api.semanticscholar.org/CorpusID:267547785. 8, 56, 58, 64, 77, 78

[14] R. Balestriero, M. Ibrahim, V. Sobal, A. S. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum. A cookbook of self-supervised learning. *ArXiv*, abs/2304.12210, 2023. 15, 16, 18

[15] P. Banerjee et al. Weakly supervised relative spatial reasoning for visual question answering. *ICCV*, 2021. 40

[16] A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *ArXiv*, abs/2105.04906, 2021. 16

[17] N. Behrmann, M. Fayyaz, J. Gall, and M. Noroozi. Long short view feature decomposition via contrastive video representation learning. In *ICCV*, 2021. 5

[18] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh. Rt-h: Action hierarchies using language. *ArXiv*, abs/2403.01823, 2024. URL https://api.semanticscholar.org/CorpusID:268249108. 86

[19] G. Bertasius, H. Wang, and L. Torresani. Is Space-Time Attention All You Need for Video Understanding? In *ICML*, page 4, 2021. 14

[20] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *ICLR*, 2020. 6

[21] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3034–3042, 2016. 8

[22] C. M. Bishop. Pattern recognition and machine learning (information science and statistics), 2006. URL https://api.semanticscholar.org/CorpusID:268095720. 72

[23] K. Black, M. Nakamoto, P. Atreya, H. R. Walke, C. Finn, A. Kumar, and S. Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *ArXiv*, abs/2310.10639, 2023. URL https://api.semanticscholar.org/CorpusID:264172455. 86

[24] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. $\pi 0$: A vision-language-action flow model for general robot control. *ArXiv*, abs/2410.24164, 2024. URL https://api.semanticscholar.org/CorpusID:273811174. 86

[25] B. Brattoli, J. Tighe, F. Zhdanov, P. Perona, and K. Chalupka. Rethinking Zero-shot Video Classification: End-to-end Training for Realistic Applications. In *CVPR*, pages 4613–4623, 2020. 22

[26] B. Brattoli, J. Tighe, F. Zhdanov, P. Perona, and K. Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *CVPR*, 2020. 5, 12

[27] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 10, 95

[28] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum,

C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. *Robotics science and systems (RSS)*, 2023. 10

[29] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020. 6, 12, 15, 16, 19, 26, 29

[30] S. Buch, C. Eyzaguirre, A. Gaidon, J. Wu, L. Fei-Fei, and J. C. Niebles. Revisiting the "video" in video-language understanding. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2907–2917, 2022. URL https://api.semanticscholar.org/CorpusID:249375461. 8

[31] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 7

[32] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 11, 12, 14, 15, 16, 18

[33] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*, 2017. 15, 20, 23, 27

[34] D. M. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil. Universal sentence encoder. *ArXiv*, 2018. 91, 100

[35] S. Changpinyo, P. K. Sharma, N. Ding, and R. Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3557–3567, 2021. URL https://api.semanticscholar.org/CorpusID:231951742. 37

[36] A. S. Chen, S. Nair, and C. Finn. Learning Generalizable Robotic Reward Functions from "In-The-Wild" Human Videos. In *Robotics: Science and Systems*, 2021. 9

[37] K. Chen, Z. Zhang, W. Zeng, R. Zhang, F. Zhu, and R. Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 7, 29, 31, 39, 40, 42, 44

[38] P. Chen, D. Huang, D. He, X. Long, R. Zeng, S. Wen, M. Tan, and C. Gan. Rspnet: Relative speed perception for unsupervised video representation learning. In *AAAI*, volume 1, 2021. 5

[39] S. Chen and D. Huang. Elaborative Rehearsal for Zero-shot Action Recognition. In *ICCV*, pages 13638–13647, 2021. 5, 18, 22

[40] S. Chen, P. Sun, Y. Song, and P. Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022. 7

[41] T. Chen, S. Saxena, L. Li, D. J. Fleet, and G. Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021. 7, 45

[42] X. Chen, Y. Li, Z. Li, Z. Wang, L. Wang, and C. Qian. Moddm: Text-to-motion synthesis using discrete diffusion model. *arXiv preprint arXiv:2308.06240*, 2023. 9

[43] F. Cheng, X. Wang, J. Lei, D. Crandall, M. Bansal, and G. Bertasius. VindLU: A Recipe for Effective Video-and-Language Pretraining. *arXiv preprint arXiv:2212.05051*, 2022. 6, 11, 12

[44] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *ArXiv*, abs/2303.04137, 2023. 9, 89

[45] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`. 6, 8, 29

[46] J. Cho et al. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. *ICCV*, 2023. 7

[47] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 6, 8

[48] A. Creswell and M. Shanahan. Faithful reasoning using large language models. *ArXiv*, abs/2208.14271, 2022. URL https://api.semanticscholar.org/CorpusID:251929296. 8

[49] Y. Cui, L. Zhao, F. Liang, Y. Li, and J. Shao. Democratizing contrastive language-image pre-training: A clip benchmark of data, model, and supervision. *arXiv preprint arXiv:2203.05796*, 2022. 6

[50] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023. 41

[51] I. R. Dave, R. Gupta, M. N. Rizve, and M. Shah. TCLR: Temporal contrastive learning for video representation. *Arxiv*, 2021. 5

[52] J. Davis and A. Bobick. The representation and recognition of action using temporal templates. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2744, 1997. 8

[53] G. C. de Croon, C. de Wagter, and T. Seidl. Enhancing optical-flow-based control by learning visual appearance cues for flying robots. *Nature Machine Intelligence*, 3:33 – 41, 2021. URL https://api.semanticscholar.org/CorpusID:231655448. 9

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Arxiv*, 2018. 14

[55] R. Devon et al. Representation learning with video deep infomax. *Arxiv*, 2020. 5

[56] P. Dhariwal and A. Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Neural Information Processing Systems*, 2021. 89

[57] A. Diba, V. Sharma, R. Safdari, D. Lotfi, S. Sarfraz, R. Stiefelhagen, and L. Van Gool. Vi2clr: Video and image for visual contrastive learning of representation. In *ICCV*, 2021. 21, 27

[58] J. Ding, N. Xue, G. Xia, and D. Dai. Decoupling zero-shot semantic segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11573–11582, 2021. 6

[59] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 14

[60] Z.-Y. Dou, A. Kamath, Z. Gan, P. Zhang, J. Wang, L. Li, Z. Liu, C. Liu, Y. LeCun, N. Peng, J. Gao, and L. Wang. Coarse-to-fine vision-language pre-training with fusion in the backbone. *ArXiv*, abs/2206.07643, 2022. 6

[61] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. PaLM-E: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 10

[62] Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, and W. Grathwohl. Reduce, Reuse, Recycle: Compositional Generation with Energy-Based Diffusion Models and MCMC. In *International Conference on Machine Learning*, 2023. 9

[63] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning Universal Policies via Text-Guided Video Generation. *arXiv:2302.00111*, 2023. 9, 86, 93, 94

[64] C. Feichtenhofer, H. Fan, B. Xiong, R. Girshick, and K. He. A large-scale study on unsupervised spatiotemporal representation learning. *Arxiv*, 2021. 5

[65] C. Fellbaum. Wordnet and wordnets. In K. Brown et al., editors, *Encyclopedia of Language & Linguistics*, pages 665–670. Elsevier, 2nd edition, 2005. 24, 25

[66] C. Finn and S. Levine. Deep Visual Foresight for Planning Robot Motion. In *IEEE International Conference on Robotics and Automation*, 2017. 9

[67] T.-J. Fu, L. Li, Z. Gan, K. Lin, W. Y. Wang, L. Wang, and Z. Liu. An empirical study of end-to-end video-language transformers with masked visual modeling. *arXiv preprint arXiv:2209.01540*, 2022. 77

[68] J. Gao, T. Zhang, and C. Xu. I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. In *AAAI*, 2019. 5

[69] J. Gao, T. Zhang, and C. Xu. I Know the Relationships: Zero-Shot Action Recognition via Two-Stream Graph Convolutional Networks and Knowledge Graphs. In *AAAI*, pages 8303–8311, 2019. 22, 23

[70] J. Gao, T. Zhang, and C. Xu. Learning to model relationships for zero-shot video classification. *TPAMI*, 2020. 5

[71] S. Ge, T. Hayes, H. Yang, X. Yin, G. Pang, D. Jacobs, J.-B. Huang, and D. Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer, 2022. 9

[72] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin. Open-vocabulary image segmentation. In *ECCV*, 2022. 6

[73] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 7

[74] Gokhale et al. Benchmarking spatial relationships in text-to-image generation, 2022. 7

[75] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *ICCV*, 2015. 5

[76] K. G. Götz. Flight control in drosophila by visual perception of motion. *Kybernetik*, 4:199–208, 1968. URL `https://api.semanticscholar.org/CorpusID:24070951`. 9

[77] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fründ, P. N. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "something something" video database for learning and evaluating visual common sense. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017. URL `https://api.semanticscholar.org/CorpusID:834612`. 69, 75

[78] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 15

[79] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. *ICLR*, 2021. 6

[80] X. Gu, C. Wen, J. Song, and Y. Gao. Seer: Language instructed video prediction with latent diffusion models. *ArXiv*, abs/2303.14897, 2023. URL https://api.semanticscholar.org/CorpusID:257766959. 86

[81] T. Gupta and A. Kembhavi. Visual programming: Compositional visual reasoning without training. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962, 2022. URL https://api.semanticscholar.org/CorpusID:253734854. 6

[82] T. Han, W. Xie, and A. Zisserman. Video representation learning by dense predictive coding. In *ICCV*, 2019. 5, 21

[83] T. Han, W. Xie, and A. Zisserman. Self-supervised co-training for video representation learning. *ArXiv*, abs/2010.09709, 2020. URL https://api.semanticscholar.org/CorpusID:224703413. 9

[84] T. Han, W. Xie, and A. Zisserman. Self-supervised Co-training for Video Representation Learning. *NeurIPS*, 33:5679–5690, 2020. 5, 21

[85] L. Hanu, J. Thewlis, Y. M. Asano, and C. Rupprecht. Vtc: Improving video-text retrieval with user comments. *ArXiv*, 2022. 9

[86] L. Hanu, A. L. Vero, and J. Thewlis. Language as the medium: Multimodal video classification through text only. *ArXiv*, abs/2309.10783, 2023. URL https://api.semanticscholar.org/CorpusID:262054213. 6, 9, 59, 66, 72

[87] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015. URL https://api.semanticscholar.org/CorpusID:1710722. 37

[88] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 9, 91, 100

[89] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video Diffusion Models. In *Neural Information Processing Systems*, 2022. 9, 90

[90] P. Hosseini, D. A. Broniatowski, and M. Diab. Knowledge-augmented language models for cause-effect relation classification. In *Proceedings of the First Workshop on Commonsense Representation and Reasoning (CSRR 2022)*, pages 43–48, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.csrr-1.6. URL https://aclanthology.org/2022.csrr-1.6. 8

[91] J. Hsu et al. What's left? concept grounding with logic-enhanced foundation models. *NeurIPS*, 2023. 7

[92] K. Hu, J. Shao, Y. Liu, B. Raj, M. Savvides, and Z. Shen. Contrast and order representations for video self-supervised learning. In *ICCV*, 2021. 21, 27

[93] Y. Hu, Y. Guo, P. Wang, X. Chen, Y.-J. Wang, J. Zhang, K. Sreenath, C. Lu, and J. Chen. Video prediction policy: A generalist robot policy with predictive visual representations. *ArXiv*, abs/2412.14803, 2024. 9, 10

[94] Y. Hu, Y. Zhang, Y. Song, Y. Deng, F. Yu, L. Zhang, W. Lin, D. Zou, and W. Yu. Seeing through pixel motion: Learning obstacle avoidance from optical flow with one camera. *ArXiv*, abs/2411.04413, 2024. URL https://api.semanticscholar.org/CorpusID:273877940. 9

[95] D. Huang, W. Wu, W. Hu, X. Liu, D. He, Z. Wu, X. Wu, M. Tan, and E. Ding. Ascnet: Self-supervised video representation learning with appearance-speed consistency. In *ICCV*, 2021. 5

[96] W. Huang, C. Wang, Y. Li, R. Zhang, and F.-F. Li. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *ArXiv*, 2024. 87

[97] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019. URL https://api.semanticscholar.org/CorpusID:152282269. 50

[98] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. R. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. π0.5: a vision-language-action model with

open-world generalization, 2025. URL https://api.semanticscholar.org/CorpusID:277993634. 86

[99] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Learning visual groups from co-occurrences in space and time. In *ICLR*, 2016. 5

[100] M. Jain, J. C. Van Gemert, T. Mensink, and C. G. Snoek. Objects2action: Classifying and localizing actions without any video example. In *ICCV*, 2015. 5

[101] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine. Planning with Diffusion for Flexible Behavior Synthesis. In *International Conference on Machine Learning*, 2022. 9

[102] Y. Jeong, J. Chun, S. Cha, and T. Kim. Object-centric world model for language-guided manipulation. *ArXiv*, abs/2503.06170, 2025. URL https://api.semanticscholar.org/CorpusID:276903201. 10

[103] 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 *ICML*, pages 4904–4916. PMLR, 2021. 5

[104] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. 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*, 2021. URL https://api.semanticscholar.org/CorpusID:231879586. 29

[105] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL https://api.semanticscholar.org/CorpusID:263830494. 8

[106] C. Ju, T. Han, K. Zheng, Y. Zhang, and W. Xie. Prompting Visual-Language Models for Efficient Video Understanding. In *ECCV*, pages 105–124. Springer, 2022. 6, 11, 12

[107] J. Kahana, N. Cohen, and Y. Hoshen. Improving zero-shot models with label distribution priors. *ArXiv*, abs/2212.00784, 2022. 6

[108] K. Kahatapitiya, A. Arnab, A. Nagrani, and M. S. Ryoo. Victr: Video-conditioned text representations for activity recognition. *ArXiv*, abs/2304.02560, 2023. 5, 6, 22, 23

[109] K. Kahatapitiya, K. Ranasinghe, J. Park, and M. S. Ryoo. Language repository for long video understanding. *ArXiv*, 2024. 69, 74, 76, 77

[110] D. Kahneman. Thinking, fast and slow, 2011. 86

[111] A. Kamath, M. Singh, Y. LeCun, G. Synnaeve, I. Misra, and N. Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. 6

[112] A. Kamath et al. What's "up" with vision-language models? investigating their struggle with spatial reasoning. *EMNLP*, 2023. 7

[113] S. Kazemzadeh, V. Ordonez, M. Matten, and T. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 42

[114] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 10

[115] S. Kim, J.-H. Kim, J. Lee, and M. Seo. Semi-parametric video-grounded text generation. *arXiv preprint arXiv:2301.11507*, 2023. 78

[116] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 20

[117] D. Ko, J. S. Lee, W. Kang, B. Roh, and H. J. Kim. Large language models are temporal and causal reasoners for video question answering. *arXiv preprint arXiv:2310.15747*, 2023. 78

[118] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. Tenenbaum. Learning to act from actionless videos through dense correspondences. *ArXiv*, abs/2310.08576, 2023. 9, 10, 86, 87, 92, 93, 94, 95, 98, 101

[119] M. Koroglu, H. Caselles-Dupr'e, G. J. Sanmiguel, and M. Cord. Onlyflow: Optical flow based motion conditioning for video diffusion models, 2024. 87

[120] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *ICCV*, pages 2556–2563. IEEE, 2011. 15, 20, 21, 22, 23, 24

[121] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu. Multi-concept customization of text-to-image diffusion, 2023. 9

[122] T. Kurutach, A. Tamar, G. Yang, S. J. Russell, and P. Abbeel. Learning Plannable Representations with Causal InfoGAN. In *Neural Information Processing Systems*, 2018. 9

[123] E. Kıcıman, R. O. Ness, A. Sharma, and C. Tan. Causal reasoning and large language models: Opening a new frontier for causality. *ArXiv*, abs/2305.00050, 2023. URL https://api.semanticscholar.org/CorpusID:258426662. 8

[124] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 7, 31

[125] J. Lee and M. S. Ryoo. Learning Robot Activities from First-Person Human Videos Using Convolutional Future Regression. In *CVPRW*, 2017. 9

[126] K. Lee, J. Gibson, and E. A. Theodorou. Aggressive perception-aware navigation using deep optical flow dynamics and pixelmpc. *IEEE Robotics and Automation Letters*, 5:1207–1214, 2020. URL https://api.semanticscholar.org/CorpusID:210064565. 9

[127] J. Lei, L. Yu, M. Bansal, and T. Berg. TVQA: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 8

[128] J. Lei, L. Yu, T. Berg, and M. Bansal. TVQA+: Spatio-temporal grounding for video question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8211–8225, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.730. URL https://aclanthology.org/2020.acl-main.730. 8

[129] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl. Language-driven semantic segmentation. *ICLR*, 2022. 6

[130] J. Li, S. Savarese, and S. C. H. Hoi. Masked unsupervised self-training for zero-shot image classification. *ArXiv*, abs/2206.02967, 2022. 6, 23

[131] J. Li, G. Shakhnarovich, and R. A. Yeh. Adapting clip for phrase localization without further training. *ArXiv*, abs/2204.03647, 2022. 6

[132] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 6, 7, 29, 30, 31, 38, 39, 41, 42, 50

[133] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 78

[134] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 41, 83

[135] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, Y. Liu, Z. Wang, J. Xu, G. Chen, P. Luo, L. Wang, and Y. Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22195–22206, 2023. URL https://api.semanticscholar.org/CorpusID:265466214. 83

[136] L. Li, J. Xu, Q. Dong, C. Zheng, Q. Liu, L. Kong, and X. Sun. Can language models understand physical concepts?, 2023. 8

[137] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao. Grounded language-image pre-training. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10955–10965, 2021. 6

[138] X. Li, C. Mata, J. Park, K. Kahatapitiya, Y. S. Jang, J. Shang, K. Ranasinghe, R. Burgert, M. Cai, Y. J. Lee, and M. S. Ryoo. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*, 2024. 8, 9

[139] X. Li, C. Mata, J. S. Park, K. Kahatapitiya, Y. S. Jang, J. Shang, K. Ranasinghe, R. Burgert, M. Cai, Y. J. Lee, and M. S. Ryoo. Llara: Supercharging robot learning data for vision-language policy. *ArXiv*, abs/2406.20095, 2024. 10, 92, 94, 95, 96, 101

[140] Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, and J.-R. Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 41, 43, 48

[141] Y. Li, C. Wang, and J. Jia. Llama-vid: An image is worth 2 tokens in large language models. *ArXiv*, abs/2311.17043, 2023. URL https://api.semanticscholar.org/CorpusID:265466723. 83

[142] J. Liang, Y. Fan, K. Zhang, R. Timofte, L. van Gool, and R. Ranjan. Movideo: Motion-aware video generation with diffusion model. In *European Conference on Computer Vision*, 2024. URL https://api.semanticscholar.org/CorpusID:273232410. 87, 99

[143] B. Lin, B. Zhu, Y. Ye, M. Ning, P. Jin, and L. Yuan. Video-llava: Learning united visual representation by alignment before projection. *ArXiv*, abs/2311.10122, 2023. URL https://api.semanticscholar.org/CorpusID:265281544. 83

[144] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 37, 48, 49

[145] W. Lin, L. Karlinsky, N. Shvetsova, H. Possegger, M. Kozinski, R. Panda, R. Feris, H. Kuehne, and H. Bischof. Match, expand and improve: Unsupervised finetuning for zero-shot action recognition with language knowledge, 2023. 9, 75, 84

[146] W. Lin, L. Karlinsky, N. Shvetsova, H. Possegger, M. Koziński, R. Panda, R. S. Feris, H. Kuehne, and H. Bischof. Match, expand and improve: Unsupervised finetuning for zero-shot action recognition with language knowledge. *ArXiv*, abs/2303.08914, 2023. URL https://api.semanticscholar.org/CorpusID:257557275. 6

[147] Y. Lin, X. Guo, and Y. Lu. Self-supervised video representation learning with meta-contrastive network. In *ICCV*, 2021. 21

[148] 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. *arXiv preprint arXiv:2208.03550*, 2022. 5

[149] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning, 2023. 15, 19, 68, 76, 85

[150] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 6, 7, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 42, 43, 44, 47, 50

[151] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, 2011. 5

[152] P. Liu, M. Lyu, I. King, and J. Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019. 89

[153] W. Liu, Y. Du, T. Hermans, S. Chernova, and C. Paxton. StructDiffusion: Language-Guided Creation of Physically-Valid Structures using Unseen Objects. In *Robotics: Science and Systems*, 2023. 9

[154] X. Liu, D. Yin, C. Zhang, Y. Feng, and D. Zhao. The magic of if: Investigating causal reasoning abilities in large language models of code. In *Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://api.semanticscholar.org/CorpusID:258968140. 8

[155] I. Loshchilov and F. Hutter. Decoupled Weight Decay Regularization. *ICLR*, 2019. 20

[156] A. Luo, X. Li, F. Yang, J. Liu, H. Fan, and S. Liu. Flowdiffuser: Advancing optical flow estimation with diffusion models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19167–19176, 2024. 10, 89

[157] H. Luo, J. Bao, Y. Wu, X. He, and T. Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. *ArXiv*, abs/2211.14813, 2022. 6, 29

[158] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, and T. Li. CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval. *Neurocomputing*, 508:293–304, 2022. 5

[159] F. Ma, X. Jin, H. Wang, Y. Xian, J. Feng, and Y. Yang. Vista-llama: Reliable video narrator via equal distance to visual tokens. *ArXiv*, abs/2312.08870, 2023. URL https://api.semanticscholar.org/CorpusID:266209773. 83

[160] M. Maaz, H. A. Rasheed, S. H. Khan, and F. S. Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Annual Meeting of the Association for Computational Linguistics*, 2023. 58, 74, 83

[161] J. Malik. Visual grouping and object recognition. In *Proceedings 11th International Conference on Image Analysis and Processing*, pages 612–621. IEEE, 2001. 7

[162] K. Mangalam, R. Akshulakov, and J. Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *ArXiv*, abs/2308.09126, 2023. URL https://api.semanticscholar.org/CorpusID:261031047. 8, 55, 64, 66, 68, 75, 76, 77

[163] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 1982. 29

[164] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016. 5

[165] S. Menon and C. Vondrick. Visual Classification via Description from Large Language Models. *arXiv preprint arXiv:2210.07183*, 2022. 5

[166] G. A. Miller. Wordnet: a lexical database for english. In *Communications of the ACM*, pages 39–41. ACM, 1995. 24, 25

[167] J. Min, S. Buch, A. Nagrani, M. Cho, and C. Schmid. Morevqa: Exploring modular reasoning models for video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13235–13245, 2024. 8, 56, 59, 64, 74, 77, 78

[168] M. Minderer, A. A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby. Simple open-vocabulary object detection with vision transformers. *ArXiv*, abs/2205.06230, 2022. URL https://api.semanticscholar.org/CorpusID:248721818. 61, 68

[169] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016. 5

[170] L. Momeni, M. Caron, A. Nagrani, A. Zisserman, and C. Schmid. Verbs in action: Improving verb understanding in video-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15579–15591, 2023. 78

[171] S. K. Muhammad Maaz, Hanoona Rasheed and F. Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *ArXiv 2306.05424*, 2023. 6, 7, 36, 37, 39, 40, 41, 42, 48, 49, 50

[172] J. Mukhoti, T.-Y. Lin, O. Poursaeed, R. Wang, A. Shah, P. H. S. Torr, and S. N. Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. *CVPR*, abs/2212.04994, 2023. 6, 29

[173] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3M: A Universal Visual Representation for Robot Manipulation. In *Conference on Robot Learning*, 2022. 9, 86

[174] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Barnes, and A. S. Mian. A comprehensive overview of large language models. *ArXiv*, abs/2307.06435, 2023. URL `https://api.semanticscholar.org/CorpusID:259847443`. 12

[175] 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*, 2022. 5, 11, 22

[176] D. Niu, Y. Sharma, G. Biamby, J. Quenum, Y. Bai, B. Shi, T. Darrell, and R. Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024. 10

[177] Nvidia, J. Bjorck, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *ArXiv*, abs/2503.14734, 2025. 86

[178] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Robotics science and systems (RSS)*, Delft, Netherlands, 2024. 10

[179] Open-X-Embodiment-Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, A. Raffin,

A. Wahid, B. Burgess-Limerick, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023. 64, 65, 69

[180] OpenAI. Gpt-4 technical report, 2023. 95

[181] OpenAI. GPT-4 technical report. https://arxiv.org/abs/2303.08774, 2023. 6, 8, 29

[182] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 10, 91, 93, 100

[183] T. Pan, Y. Song, T. Yang, W. Jiang, and W. Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *CVPR*, 2021. 21

[184] P. Papalampidi, S. Koppula, S. Pathak, J. Chiu, J. Heyward, V. Patraucean, J. Shen, A. Miech, A. Zisserman, and A. Nematzdeh. A simple recipe for contrastively pre-training video-first encoders beyond 16 frames. *arXiv preprint arXiv:2312.07395*, 2023. 8, 58, 64, 77

[185] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto. The surprising effectiveness of representation learning for visual imitation, 2021. 9

[186] J. S. Park, K. Ranasinghe, K. Kahatapitiya, W. Ryoo, D. Kim, and M. S. Ryoo. Too many frames, not all useful: Efficient strategies for long-form video qa. *ArXiv*, abs/2406.09396, 2024. URL https://api.semanticscholar.org/CorpusID:270440923. 64, 69, 74, 76, 77, 78

[187] V. Pătrăucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *ICLR (Workshop)*, 2016. 5

[188] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 7, 31, 42, 44

[189] A. Piergiovanni, A. Angelova, and M. S. Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020. 5

[190] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui. Spatiotemporal contrastive video representation learning. *CVPR*, 2021. 5, 21, 27

[191] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui. Spatiotemporal Contrastive Video Representation Learning. In *CVPR*, pages 6964–6974, 2021. 20

[192] R. Qian, Y. Li, Z. Xu, M.-H. Yang, S. Belongie, and Y. Cui. Multimodal Open-Vocabulary Video Classification via Pre-Trained Vision and Language Models. *arXiv preprint arXiv:2207.07646*, 2022. 6, 11, 12, 23

[193] J. Qin, L. Liu, L. Shao, F. Shen, B. Ni, J. Chen, and Y. Wang. Zero-Shot Action Recognition with Error-Correcting Output Codes. In *CVPR*, pages 2833–2842, 2017. 22

[194] A. Radford and K. Narasimhan. Improving language understanding by generative pre-training, 2018. URL `https://api.semanticscholar.org/CorpusID:49313245`. 72

[195] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019. 91

[196] 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 *ICML*, pages 8748–8763. PMLR, 2021. 5, 11, 12, 13, 14, 15, 21, 22, 23, 24, 26, 27, 84

[197] 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*, pages 8748–8763. PMLR, 2021. 6, 29, 32

[198] S. Ramasinghe, J. Rajasegaran, V. Jayasundara, K. Ranasinghe, R. Rodrigo, and A. A. Pasqual. Combined static and motion features for deep-networks-based activity recognition in videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 29:2693–2707, 2018. URL `https://api.semanticscholar.org/CorpusID:53116615`. 8

[199] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *preprint*, 2022. [arxiv:2204.06125]. 9, 89

[200] K. Ranasinghe and M. S. Ryoo. Language-based action concept spaces improve video self-supervised learning. In *NeurIPS*, 2023. 6, 9

[201] K. Ranasinghe, M. Naseer, S. H. Khan, F. S. Khan, and M. S. Ryoo. Self-supervised video transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2864–2874, 2021. 5, 11, 12, 14, 15, 16, 18, 20, 21, 24, 27

[202] K. Ranasinghe, M. Naseer, S. H. Khan, F. S. Khan, and M. S. Ryoo. Self-supervised video transformer. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2864–2874, 2021. URL https://api.semanticscholar.org/CorpusID:244800737. 8

[203] K. Ranasinghe, B. McKinzie, S. Ravi, Y. Yang, A. Toshev, and J. Shlens. Perceptual grouping in contrastive vision-language models. In *ICCV*, 2023. 6, 29

[204] K. Ranasinghe, B. McKinzie, S. Ravi, Y. Yang, A. Toshev, and J. Shlens. Perceptual grouping in contrastive vision-language models. *ICCV*, 2023. 19, 26

[205] K. Ranasinghe, X. Li, K. Kahatapitiya, and M. S. Ryoo. Understanding long videos in one multimodal language model pass, 2024. 7

[206] K. Ranasinghe, S. N. Shukla, O. Poursaeed, M. S. Ryoo, and T.-Y. Lin. Learning to localize objects improves spatial reasoning in visual-llms. In *CVPR*, 2024. 62, 67, 83

[207] H. Rasheed, M. U. Khattak, M. Maaz, S. Khan, and F. S. Khan. Fine-tuned clip models are efficient video learners. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6545–6554, 2022. URL https://api.semanticscholar.org/CorpusID:254366626. 6, 22

[208] H. Rasheed, M. U. Khattak, M. Maaz, S. Khan, and F. S. Khan. Fine-tuned CLIP Models are Efficient Video Learners. *arXiv preprint arXiv:2212.03640*, 2022. 6, 11, 12

[209] A. Recasens, P. Luc, J.-B. Alayrac, L. Wang, F. Strub, C. Tallec, M. Malinowski, V. Pătrăucean, F. Altché, M. Valko, et al. Broaden Your Views for Self-Supervised Video Learning. In *ICCV*, pages 1255–1265, 2021. 22

[210] A. Recasens, P. Luc, J.-B. Alayrac, L. Wang, F. Strub, C. Tallec, M. Malinowski, V. Patraucean, F. Altché, M. Valko, et al. Broaden your views for self-supervised video learning. *ICCV*, 2021. 5, 21, 27

[211] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 7

[212] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and F. de Nando. A generalist agent. In *Trans. on Machine Learning Research*, 2022. 10

[213] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. *ArXiv*, abs/2501.06994, 2025. URL https://api.semanticscholar.org/CorpusID:275471722. 9, 86

[214] Z. Ren, Z. Pan, X. Zhou, and L. Kang. Diffusion motion: Generate text-guided 3d human motion by diffusion model. *arXiv preprint arXiv:2210.12315*, 2022. 9

[215] J. Robinson, C. Rytting, and D. Wingate. Leveraging large language models for multiple choice question answering. *ICLR*, 2023. 56, 58, 66, 71, 72

[216] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 90

[217] I. G. Ros and A. A. Biewener. Optic flow stabilizes flight in ruby-throated hummingbirds. *Journal of Experimental Biology*, 219:2443 – 2448, 2016. URL https://api.semanticscholar.org/CorpusID:11106817. 9

[218] M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1709–1718, 2006. URL https://api.semanticscholar.org/CorpusID:14039104. 11, 40

[219] M. Safaei and H. Foroosh. Still image action recognition by predicting spatial-temporal pixel evolution. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 8

[220] S. Saxena, C. Herrmann, J. Hur, A. Kar, M. Norouzi, D. Sun, and D. J. Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. *ArXiv*, abs/2306.01923, 2023. 10, 90, 91

[221] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. 6

[222] M. C. Schiappa, Y. S. Rawat, and M. Shah. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 2022. 11

[223] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *ArXiv*, abs/1508.07909, 2015. URL https://api.semanticscholar.org/CorpusID:1114678. 75

[224] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg. Concept2Robot: Learning Manipulation Concepts from Instructions and Human Demonstrations. *IJRR*, 2021. 9

[225] P. Sharma, D. Pathak, and A. Gupta. Third-person visual imitation learning via decoupled hierarchical controller. In *Neural Information Processing Systems*, 2019. 9

[226] Y. Sharma, Y. Zhu, C. Russell, and T. Brox. Pixel-level correspondence for self-supervised learning from video. *ArXiv*, abs/2207.03866, 2022. URL https://api.semanticscholar.org/CorpusID:250407930. 9

[227] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos, 2025. 87

[228] L. X. Shi, B. Ichter, M. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang, N. Fusai, A. Li-Bell, D. Driess, L. Groom, S. Levine, and C. Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *ArXiv*, abs/2502.19417, 2025. URL https://api.semanticscholar.org/CorpusID:276618098. 86

[229] M. Shridhar, Y. L. Lo, and S. James. Generative image as action models. *ArXiv*, abs/2407.07875, 2024. 9, 87

[230] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman. Make-a-video: Text-to-video generation without text-video data, 2022. 9

[231] A. Sivakumar, K. Shaw, and D. Pathak. Robotic Telekinesis: Learning a Robotic Hand Imitator by Watching Humans on Youtube. In *Robotics: Science and Systems*, 2022. 9

[232] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 6

[233] K. Song, B. Chen, M. Simchowitz, Y. Du, R. Tedrake, and V. Sitzmann. History-guided video diffusion, 2025. 91

[234] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *Arxiv*, 2012. 15, 20

[235] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv preprint arXiv:1212.0402*, 2012. 21, 22, 23

[236] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015. 5

[237] H.-T. Su, Y. Niu, X. Lin, W. H. Hsu, and S.-F. Chang. Language models are causal knowledge extractors for zero-shot video question answering. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4951–4960, 2023. URL https://api.semanticscholar.org/CorpusID:258041332. 78

[238] S. Sudhakar, R. Liu, B. V. Hoorick, C. Vondrick, and R. Zemel. Controlling the world by sleight of hand. *ArXiv*, abs/2408.07147, 2024. 9, 10, 87

[239] S.-H. Sun, H. Noh, S. Somasundaram, and J. Lim. Neural program synthesis from diverse demonstration videos. In *International Conference on Machine Learning*, 2018. 9

[240] D. Sur'is, S. Menon, and C. Vondrick. Vipergpt: Visual inference via python execution for reasoning. *ArXiv*, abs/2303.08128, 2023. URL https://api.semanticscholar.org/CorpusID:257505358. 6

[241] D. Surís, S. Menon, and C. Vondrick. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*, 2023. 78

[242] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, pages 1195–1204, 2017. 6

[243] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 8

[244] Z. Teed and J. Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020. 86, 89, 90

[245] Y. Tian, S. Yang, J. Zeng, P. Wang, D. Lin, H. Dong, and J. Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. *ArXiv*, abs/2412.15109, 2024. URL `https://api.semanticscholar.org/CorpusID:274859727`. 10

[246] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 7

[247] Z. Tong, Y. Song, J. Wang, and L. Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *ArXiv*, abs/2203.12602, 2022. 5, 11, 12, 21, 27

[248] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 6, 29, 32

[249] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 6, 8, 29, 33

[250] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 7

[251] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, H. Moraldo, H. Zhang, M. T. Saffar, S. Castro, J. Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual description, 2022. 9

[252] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 5

[253] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018.

[254] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 5

[255] H.-C. Wang, S.-F. Chen, and S.-H. Sun. Diffusion Model-Augmented Behavioral Cloning. *arXiv:2302.13335*, 2023. 9

[256] J. Wang, L. Yuan, Y. Zhang, and H. Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024. 8, 77

[257] M. Wang, J. Xing, and Y. Liu. ActionCLIP: A New Paradigm for Video Action Recognition. *arXiv preprint arXiv:2109.08472*, 2021. 5, 11, 12, 22

[258] Q. Wang and K. Chen. Alternative semantic representations for zero-shot human action recognition. In *ECML/PKDD*, 2017. 9

[259] Q. Wang and K. Chen. Alternative Semantic Representations for Zero-Shot Human Action Recognition. In *ECML-PKDD*, pages 87–102. Springer, 2017. 22

[260] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr. Fast online object tracking and segmentation: A unifying approach. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1328–1338, 2018. URL `https://api.semanticscholar.org/CorpusID:54475412`. 61

[261] S. Wang, Q. Zhao, M. Q. Do, N. Agarwal, K. Lee, and C. Sun. Vamos: Versatile action models for video understanding. *arXiv preprint arXiv:2311.13627*, 2023. 8, 64, 69, 77, 78

[262] W. Wang, Z. Chen, X. Chen, J. Wu, X. Zhu, G. Zeng, P. Luo, T. Lu, J. Zhou, Y. Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023. 7, 31, 45

[263] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 5

[264] X. Wang, Y. Zhang, O. Zohar, and S. Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2025. 56, 64, 74, 77, 78

[265] Y. Wang and Y. Zhao. Gemini in reasoning: Unveiling commonsense in multimodal large language models. *ArXiv*, abs/2312.17661, 2023. URL https://api.semanticscholar.org/CorpusID:266690844. 8, 55

[266] Y. Wang, K. Li, Y. Li, Y. He, B. Huang, Z. Zhao, H. Zhang, J. Xu, Y. Liu, Z. Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 58, 77, 78

[267] Y. Wang, Y. Yang, and M. Ren. Lifelongmemory: Leveraging llms for answering queries in long-form egocentric videos. In *arxiv*, 2023. 8, 74, 77

[268] Y. Wang, K. Li, X. Li, J. Yu, Y. He, G. Chen, B. Pei, R. Zheng, J. Xu, Z. Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *arXiv e-prints*, pages arXiv–2403, 2024. 8, 77

[269] Z. Wang, S. Yu, E. Stengel-Eskin, J. Yoon, F. Cheng, G. Bertasius, and M. Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*, 2024. 8, 74, 76, 77, 78

[270] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652, 2021. URL https://api.semanticscholar.org/CorpusID:237416585. 7

[271] J. Weston and S. Sukhbaatar. System 2 attention (is something you might need too). *ArXiv*, abs/2311.11829, 2023. URL https://api.semanticscholar.org/CorpusID:265295357. 8

[272] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023. 10

[273] H. Wu, D. Li, B. Chen, and J. Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024. URL https://arxiv.org/abs/2407.15754. 74

[274] W. Wu, Z. Sun, and W. Ouyang. Revisiting Classifier: Transferring Vision-Language Models for Video Recognition. *AAAI*, 2023. 14, 16

[275] F. Xiao, J. Tighe, and D. Modolo. Modist: Motion distillation for self-supervised video representation learning. *Arxiv*, 2021. 21, 22

[276] J. Xiao, X. Shang, A. Yao, and T.-S. Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8, 55, 64, 68, 78

[277] J. Xiao, A. Yao, Z. Liu, Y. Li, W. Ji, and T.-S. Chua. Video as conditional graph hierarchy for multi-granular question answering. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2804–2812, 2022. 8

[278] J. Xiao, P. Zhou, T.-S. Chua, and S. Yan. Video graph transformer for video question answering. In *European Conference on Computer Vision*, pages 39–58. Springer, 2022. 8

[279] J. Xiao, P. Zhou, A. Yao, Y. Li, R. Hong, S. Yan, and T.-S. Chua. Contrastive video question answering via video graph transformer. *arXiv preprint arXiv:2302.13668*, 2023. 78

[280] J. Xiao, A. Yao, Y. Li, and T.-S. Chua. Can i trust your answer? visually grounded video question answering. *CVPR*, 2024. 55

[281] D. Xu, Z. Zhao, J. Xiao, F. Wu, H. Zhang, X. He, and Y. Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017. 8

[282] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao. GMFlow: Learning Optical Flow via Global Matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 89

[283] H. Xu, L. Han, Q. Yang, M. Li, and M. Srivastava. Penetrative ai: Making llms comprehend the physical world, 2024. 8

[284] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, and X. Wang. GroupViT: Semantic Segmentation Emerges from Text Supervision. In *CVPR*, pages 18134–18144, 2022. 6, 26

[285] J. Xu, J. Hou, Y. Zhang, R. Feng, Y. Wang, Y. Qiao, and W. Xie. Learning open-vocabulary semantic segmentation models from natural language supervision. *ArXiv*, 2023. 6, 26

[286] X. Xu, T. M. Hospedales, and S. Gong. Multi-Task Zero-Shot Action Recognition with Prioritised Data Augmentation . In *ECCV*, pages 343–359. Springer, 2016. 22

[287] X. Xu, T. Hospedales, and S. Gong. Transductive zero-shot action recognition by word-vector embedding. *IJCV*, 2017. 5, 12

[288] H. Xue, Y. Sun, B. Liu, J. Fu, R. Song, H. Li, and J. Luo. CLIP-ViP: Adapting Pre-trained Image-Text Model to Video-Language Representation Alignment. *arXiv preprint arXiv:2209.06430*, 2022. 6, 11, 12

[289] S. Yan, T. Zhu, Z. Wang, Y. Cao, M. Zhang, S. Ghosh, Y. Wu, and J. Yu. Videococa: Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv*, 2022. URL `https://api.semanticscholar.org/CorpusID:254535696`. 41, 83

[290] S. Yan, T. Zhu, Z. Wang, Y. Cao, M. Zhang, S. Ghosh, Y. Wu, and J. Yu. Video-Text Modeling with Zero-Shot Transfer from Contrastive Captioners. *arXiv preprint arXiv:2212.04979*, 2022. 6, 11, 12

[291] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1686–1697, 2021. 41, 78, 83

[292] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 35:124–141, 2022. 77, 83

[293] A. Yang, A. Miech, J. Sivic, I. Laptev, and C. Schmid. Zero-shot video question answering via frozen bidirectional language models. *arXiv preprint arXiv:2206.08155*, 2022. 41

[294] L. Yao, R. Huang, L. Hou, G. Lu, M. Niu, H. Xu, X. Liang, Z. Li, X. Jiang, and C. Xu. FILIP: Fine-grained interactive language-image pre-training. In *ICLR*, 2022. 6

[295] Q. Ye, G. Xu, M. Yan, H. Xu, Q. Qian, J. Zhang, and F. Huang. Hitea: Hierarchical temporal-aware video-language pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15405–15416, 2023. 78

[296] Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 77

[297] H. You, H. Zhang, Z. Gan, X. Du, B. Zhang, Z. Wang, L. Cao, S.-F. Chang, and Y. Yang. Ferret: Refer and ground anything anywhere at any granularity. *ArXiv*, abs/2310.07704, 2023. URL https://api.semanticscholar.org/CorpusID:263834718. 7, 31

[298] J. Yu, X. Wang, S. Tu, S. Cao, D. Zhang-Li, X. Lv, H. Peng, Z. Yao, X. Zhang, H. Li, C. Li, Z. Zhang, Y. Bai, Y. Liu, A. Xin, N. Lin, K. Yun, L. Gong, J. Chen, Z. Wu, Y. Qi, W. Li, Y. Guan, K. Zeng, J. Qi, H. Jin, J. Liu, Y. Gu, Y. Yao, N. Ding, L. Hou, Z. Liu, B. Xu, J. Tang, and J. Li. Kola: Carefully benchmarking world knowledge of large language models, 2023. 8, 55

[299] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3521–3529, 2016. URL https://api.semanticscholar.org/CorpusID:10132533. 44

[300] S. Yu, J. Cho, P. Yadav, and M. Bansal. Self-chained image-language model for video localization and question answering. *ArXiv*, abs/2305.06988, 2023. URL https://api.semanticscholar.org/CorpusID:258615748. 8

[301] S. Yu, J. Cho, P. Yadav, and M. Bansal. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36, 2024. 65, 77, 78

[302] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. In *Conference on Robot Learning*, 2019. 93

[303] Z. Yu, D. Xu, J. Yu, T. Yu, Z. Zhao, Y. Zhuang, and D. Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. *ArXiv*, abs/1906.02467, 2019. URL https://api.semanticscholar.org/CorpusID:69645185. 8, 74, 83

[304] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024. 10

[305] S. Yun, J. Kim, D. Han, H. Song, J.-W. Ha, and J. Shin. Time is MattEr: Temporal self-supervision for video transformers. In *Proceedings of the 39th International Conference on Machine Learning*, pages 25804–25816, 2022. 8, 55

[306] Y. Zang, W. Li, J. Han, K. Zhou, and C. C. Loy. Contextual object detection with multimodal large language models. *arXiv preprint arXiv:2305.18279*, 2023. 7, 31

[307] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine. Robotic control via embodied chain-of-thought reasoning. In *Conference on Robot Learning*, 2024. URL https://api.semanticscholar.org/CorpusID:271097636. 10

[308] R. Zellers and Y. Choi. Zero-shot activity recognition with verb attribute induction. *arXiv preprint arXiv:1707.09468*, 2017. 5

[309] R. Zellers, X. Lu, J. Hessel, Y. Yu, J. S. Park, J. Cao, A. Farhadi, and Y. Choi. Merlot: Multimodal neural script knowledge models. *ArXiv*, abs/2106.02636, 2021. 21

[310] A. Zeng, A. Wong, S. Welker, K. Choromanski, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke, et al. Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language. *arXiv preprint arXiv:2204.00598*, 2022. 9, 63

[311] K.-H. Zeng, T.-H. Chen, C.-Y. Chuang, Y.-H. Liao, J. C. Niebles, and M. Sun. Leveraging video descriptions to learn video question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. doi:

10.1609/aaai.v31i1.11238. URL https://ojs.aaai.org/index.php/AAAI/article/view/11238. 8

[312] Y. Zeng, X. Zhang, and H. Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *ArXiv*, abs/2111.08276, 2021. 6

[313] Z. Zeng, Y. Ge, X. Liu, B. Chen, P. Luo, S. Xia, and Y. Ge. Learning transferable spatiotemporal representations from natural script knowledge. *ArXiv*, abs/2209.15280, 2022. 21

[314] C. Zhang, T. Lu, M. M. Islam, Z. Wang, S. Yu, M. Bansal, and G. Bertasius. A simple llm framework for long-range video question-answering. *arXiv preprint arXiv:2312.17235*, 2023. 8, 55, 56, 60, 61, 63, 64, 65, 66, 69, 70, 74, 76, 77, 78, 79, 80, 83, 84

[315] H. Zhang, P. Zhang, X. Hu, Y.-C. Chen, L. H. Li, X. Dai, L. Wang, L. Yuan, J.-N. Hwang, and J. Gao. Glipv2: Unifying localization and vision-language understanding. *ArXiv*, abs/2206.05836, 2022. 6

[316] H. Zhang, X. Li, and L. Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *ArXiv*, abs/2306.02858, 2023. URL https://api.semanticscholar.org/CorpusID:259075356. 41, 83

[317] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 9

[318] M. Zhang, Z. Cai, L. Pan, F. Hong, X. Guo, L. Yang, and Z. Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 9

[319] R. Zhang, J. Han, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, P. Gao, and Y. J. Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *ArXiv*, abs/2303.16199, 2023. URL https://api.semanticscholar.org/CorpusID:257771811. 41, 83

[320] S. Zhang, P. Sun, S. Chen, M. Xiao, W. Shao, W. Zhang, K. Chen, and P. Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023. 7, 31

[321] Y. Zhang, Z. Wang, J. H. Liew, J. Huang, M. Zhu, J. Feng, and W. Zuo. Associating spatially-consistent grouping with text-supervised semantic segmentation. *ArXiv*, abs/2304.01114, 2023. 6, 29

[322] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Nie, and J. rong Wen. A survey of large language models. *ArXiv*, abs/2303.18223, 2023. URL https://api.semanticscholar.org/CorpusID:257900969. 12

[323] Y. Zhao, I. Misra, P. Krahenbuhl, and R. Girdhar. Learning video representations from large language models. *ArXiv*, abs/2212.04501, 2022. 21, 22

[324] Y. Zhao, Z. Lin, D. Zhou, Z. Huang, J. Feng, and B. Kang. Bubogpt: Enabling visual grounding in multi-modal llms. *arXiv preprint arXiv:2307.08581*, 2023. 7, 31

[325] Z. Zhao, H. Ma, and S. You. Single image action recognition using semantic body part actions. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 8

[326] Z. Zhao, W. S. Lee, and D. Hsu. Large language models as commonsense knowledge for large-scale task planning, 2023. 8

[327] J. Zheng, J. Li, D. Liu, Y. Zheng, Z. Wang, Z. Ou, Y. Liu, J. Liu, Y.-Q. Zhang, and X. Zhan. Universal actions for enhanced embodied foundation models. *ArXiv*, abs/2501.10105, 2025. URL https://api.semanticscholar.org/CorpusID:275606605. 86

[328] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024. 10

[329] C. Zhou, C. C. Loy, and B. Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, 2021. 6

[330] Y. Zhu, Y. Long, Y. Guan, S. Newsam, and L. Shao. Towards Universal Representation for Unseen Action Recognition. In *CVPR*, pages 9436–9445, 2018. 5, 18, 22, 23

[331] X. Zou, J. Yang, H. Zhang, F. Li, L. Li, J. Gao, and Y. J. Lee. Segment everything everywhere all at once. In *NeurIPS*, 2023. 36, 48