Open-loop VLM Robot Planning: An Investigation of Fine-tuning and Prompt Engineering Strategies

Shogo Akiyama[1], Rousslan Fernand Julien Dossa[1], Kai Arulkumaran[1], Shivakanth Sujit[1] and Edward Johns[2]

***Abstract*— Recent works have suggested that language-based foundation models contain commonsense knowledge and are capable of performing basic reasoning. This has significant promise in robotics for task-level planning. As an example, the recent EgoPlan-Bench benchmark studies egocentric, embodied planning, measured through multiple-choice questions on captioned videos. In this work, we thoroughly examine the benchmark using open-source 7/13B-parameter models and investigate the impact of different sources of training data, as well as prompting strategies that are widely used outside of the robotics domain. Our experiments show that (1) in-domain and out-of-domain performance is, unsurprisingly, connected with training and evaluation dataset overlap, and (2) surprisingly, prompting strategies that have been effective in other domains, fail to significantly increase performance here.**

I. INTRODUCTION

Foundation models are large models that have been trained on a wide distribution of data, and thereby can provide a "foundation" for different use cases, ranging from dialogue agents to robot controllers [1]. By learning from large-scale data, these models can exhibit varied and even nuanced commonsense knowledge, which was infeasible with traditional symbolic AI. A seminal work in this area is GPT-3, a large language model (LLM) that not only demonstrated such knowledge but could also perform few-shot learning without further parameter adaptation, simply through text prompting [2]. Given LLM's abilities to adapt to prompts, there is now a growing research field on improving these [3].

The use of LLMs and vision-language models (VLMs) has also spread to robotics research [4]. One of the most obvious use cases is to replace symbolic task planners with planners that operate using natural language [5], [6]. Although there have been several works that use foundation models for low-level control [7], [8], [9], [10], foundation models have been shown to be particularly powerful for task planning when given access to lower-level control policies for motion control [5], [6], [11]. Impressively, Hu et al. [6] demonstrated the GPT-4V VLM [12] could perform closed-loop task planning for a real robot through prompting techniques alone.

However, the use of large, closed-source models that are either accessible only through an API, or not accessible at all, presents an issue for wide-scale deployment in robotics. Such models cannot be inspected (for interpretability and/or safety), are not private, require an internet connection, might require payment, and require large amounts of compute resources, even just for inference. Upstream changes are beyond the control of the end-user, and can cause performance degradations [13], [14]. Therefore, we believe it is important to study open-source models that can at least be deployed on consumer-grade GPUs. Furthermore, such studies should ideally use reproducible benchmarks, allowing for more rigor.

In this paper, we use the recently-released EgoPlan-Bench [15] to study the task planning capabilities of open-source VLMs. As in the original work [15], we base our study on the 7B-parameter variant of VideoLLaMA [16], a VLM which can operate over video frames. EgoPlan-Bench converts the egocentric video data from the EPIC-KITCHENS dataset [17] into a multiple-choice question-answer form to evaluate a VLM's ability to integrate past information, the current context, and commonsense knowledge, in order to solve open-loop[1] planning problems. EgoPlan-Bench also includes out-of-domain evaluation on a subset of the Ego4D dataset [18] that is processed the same way. As the task involves embodied scene reasoning, we investigate if fine-tuning VideoLLaMA with additional physical object-grounding [19] and robot planning [20] datasets can further improve performance. We also investigate three prompting methods commonly used in broader foundation model research: chain-of-thought (CoT) [21], self-verification (SV) [22], and self-consistency (SC) [23].

We find that fine-tuning on these additional datasets does not improve performance on EgoPlan-Bench, which we attribute to the dissimilarities between these datasets' features and the evaluation domains (Subsection V-C). Despite their success in other domains, surprisingly we found that the prompting methods reduced performance with fine-tuned models—though the base models were left largely unaffected. This indicates that caution is needed to preserve the reasoning abilities of models tuned for robot planning, if they reason at all (Section VI).

II. MODEL

VideoLLaMA [16] is a video VLM that augments the Vicuna LLM [24] with pre-trained visual (BLIP-2 [25]) and audio (ImageBind [26]) encoders (Figure 1). To extend the visual encoder to video, VideoLLaMA embeds multiple keyframes separately, which are then combined with positional encodings, then processed by a video Q-Former [25], [16] to obtain video embeddings. This is then

*This work was supported by JST under Moonshot R&D Grant Number JPMJMS2012.

[1]Araya, 1-11 Kanda Sakumachō, Chiyoda, Tokyo 101-0025, Japan {akiyama_shogo,dossa,kai_arulkumaran}@araya.org
[2]Imperial College London, South Kensington, London SW7 2BX, UK e.johns@imperial.ac.uk

[1]We call this setting open-loop, as question-answer pairs are evaluated independently, instead of consecutively in a trajectory.

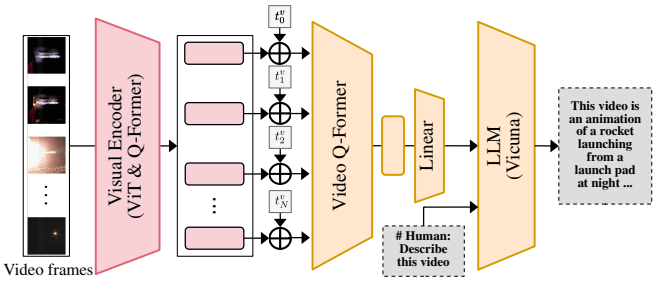

Fig. 1: Vision and language components of VideoLLaMA; we exclude the audio component as it is not used in EgoPlan-Bench. Results in EgoPlan-Bench [15] come from fine-tuning all components apart from the visual encoder; in our experiments, we additionally explored fine-tuning the latter.

adapted as input for the LLM through a linear transformation. Finally, the LLM uses the combination of the transformed video embedding and the language instruction embeddings to produce text as output. As in EgoPlan-Bench [15], we use this pre-trained model as a basis for our experiments.

III. DATASETS

In this section, we summarise the datasets used and provide some quantitative characteristics in Table I. VideoLLaMA is trained on the LLaVA-Instruct-150K [27], CC-SBU [28], [29] and InternVideo [30] datasets. In addition to the EgoPlan-IT dataset introduced in EgoPlan-Bench [15], we also experiment with the RoboVQA [20] and PhysObjects [19] datasets.

TABLE I: Dataset characteristics.

Dataset	Training QA Pairs	Answer Vocabulary Size
LLaVA-Instruct-150K	150,000	50,232
CC-SBU	3,439	2,811
InternVideo	11,189	16,823
EgoPlan-IT	50,285	291
RoboVQA	798,429	1,880
PhysObjects	456,600	42

LLaVA-Instruct-150K: This dataset was introduced with the LLaVA model [27] for visual instruction fine-tuning of VLMs. It was constructed by using the GPT-4 LLM [31] to generate instruction-following data based on captions and descriptions of objects' spatial relationships from the COCO dataset [32]. The vision-language instruction-following samples are divided into conversational formats, detailed description formats, and complex reasoning instructions.

CC-SBU: This dataset was introduced with the MiniGPT model series [28], [29] to provide curated data designed for aligning vision and language-based representations.

InternVideo: This dataset was introduced with the InternVideo model [30], with a focus on spatiotemporal reasoning, event localization, and causal relationship inference from videos. It was constructed by using a combination of ChatGPT [33] and VideoChat [34] to annotate data from the WebVid10M video dataset [35].

EgoPlan-IT: This dataset accompanies EgoPlan-Bench [15], and consists of action-centric captioning of the EPIC-KITCHENS [17] and Ego4D [18] video datasets, where the latter is only used for evaluation. The original captions are augmented by using EgoVLP [36] to segment the videos and GPT-4 to decompose goals into a series of actions. The main samples are formatted as question and multiple-choice answer pairs. Additional samples are formatted for action recognition and contrastive losses.

RoboVQA: This dataset consists of 238 hours of egocentric video data collected from multiple humans and tele-operated robots from 3 different office environments [20]. Crowd-sourcing was used to temporally segment the videos, and provide instructions on how to complete the current sub-task.

PhysObjects: This dataset consists of object-centric, visually-grounded data based on physical properties such as mass, fragility, deformability, material, transparency, etc. [19]. 417K annotations were automatically generated from the EgoObjects dataset [37] using included metadata and object detection methods. A further 39.6K annotations were crowd-sourced for scenarios that could not be automatically annotated (e.g., continuous physical properties). The authors showed that a VLM fine-tuned on such dataset could improve a planning pipeline for a real robot, as evaluated by a human.

IV. METHODS

The two main ways of adapting foundation models for downstream tasks are a) fine-tuning, and b) prompting.

A. Fine-tuning

Firstly, we investigated fine-tuning parameters within VideoLLaMA to adapt it for EgoPlan-Bench. We used the original authors' pipeline and fine-tuning hyperparameters [15]; all parameters within the video Q-Former and linear transformation are updated, whilst the LLM is fine-tuned using low-rank adaptation (LoRA) [38]. We also performed an additional experiment where we fine-tuned all of the visual encoder weights. The original EgoPlan-Bench experiments fine-tuned VideoLLaMA on a combination of the LLaVA-Instruct-150K, CC-SBU, InternVideo, and EgoPlan-IT datasets[15]. In our experiments, we add the RoboVQA and PhysObjects datasets (separately, and jointly).

B. Prompting methods

The evaluation for EgoPlan-Bench is as follows: an image from the current time step is provided, alongside preceding video keyframes, and a text prompt that gives the task for the VLM. The prompt includes a high-level task goal, explains the visual inputs, and then asks what the next action should be. The model is then evaluated 4 times, each with a different candidate action appended at the end of the prompt; the model's answer is taken as the action with the highest probability. In addition to this "base" prompt, we experimented with more sophisticated prompting methods:

Chain-of-thought: CoT encourages LLMs to perform reasoning before outputting their final answer, which was shown to improve performance in more complex decision-making

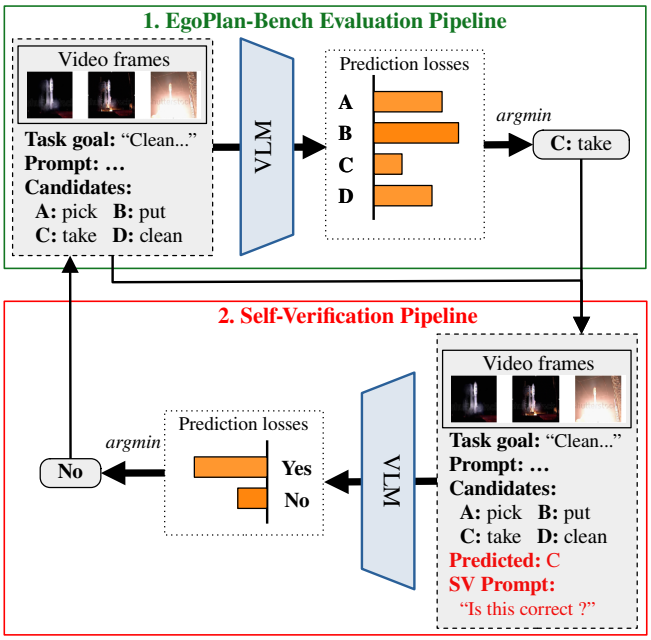

Fig. 2: Overview of the EgoPlan-Bench evaluation protocol, and our proposed SV augmentation.

tasks [21]. To implement CoT, we appended "Think step by step" [21], [39] to the end of the original EgoPlan-Bench prompts.

Self-verification: SV asks LLMs to review their own answers, as a form of introspection [22]. Later works considered using this introspection to ask LLMs to refine their answers, if warranted by the original SV response [40], [41].

We augmented the standard EgoPlan-Bench evaluation protocol with a two-step SV process based on "true-false item verification" (TFIV) [22]. First, the original EgoPlan-Bench prompt and the model's answer are reformulated as a statement that the model should verify. If the reasoning is false, the history of the conversation is fed back to the model, along with an instruction to rectify the prediction, as illustrated in Figure 2.

Self-consistency: SC is based on the idea that sampling LLMs for answers to reasoning problems (involving open-ended reasoning) can result in different answers, with the most likely answer sampled the most consistently. Thus, SC aims to improve reasoning performance by sampling multiple answers and using majority voting to pick the final answer.

We followed the original protocol [23], in which a one-shot example[2] in the form "{original prompt} {reasoning} The answer is {option}" is prepended to the prompt, and all candidate answers are given at the end of the prompt. We then generated 20 completions and parsed the outputs for an exact match for the correct answer.

V. RESULTS

All of our results on EgoPlan-Bench can be found in Table II. As an overall trend, performance did not seem to be correlated with sample frequency, i.e., the success rate with frequent or infrequent nouns/verbs was roughly the same.

A. Fine-tuning

Fine-tuning on RoboVQA did not significantly impact the scores on EPIC-KITCHENS (in-domain evaluation), but led to a slight regression on Ego4D (out-of-domain evaluation). Fine-tuning on PhysObjects did not significantly impact scores on either domain. Co-fine-tuning resulted in similar changes as to training purely on RoboVQA. To investigate further, we analyzed the similarities between the datasets we used, detailed in Subsection V-C.

We observed some improvement in performance from allowing the visual encoder component to be fine-tuned, though, as before, adding additional datasets did not help further. However, this improvement was only for in-domain evaluation, and the model did not exhibit any increased ability to perform visually-grounded planning out-of-domain. Allowing VLMs to adapt their low-level visual representations can help slightly for adaptation to a particular domain, but would not necessarily aid generalization.

Due to resource limitations, we were unable to fine-tune the 13B variant of VideoLLaMA[3], but included the results from evaluating the model with its original parameters. Notably, despite almost double the parameters, its performance was almost identical to the 7B variant—and significantly lower than that of all models trained with EgoPlan-IT. With the caveat that these are not extremely large models, in this case, scale did not improve performance.

B. Prompting methods

Across all experiments, CoT does not significantly impact scores. The most notable difference—a decrease in performance—occurs for the 7B model trained with the original EgoPlan-Bench settings (LLaVA-Instruct + CC-SBU + InternVideo + EgoPlan-IT). In conjunction with other results, we believe the VLM overfits to EgoPlan-IT's prompt format, to the detriment of its instruction-following ability.

While SV has little impact on the original VideoLLaMA models, it can decrease the scores of models trained on EgoPlan-IT by more than 10% on EPIC-KITCHENS, and 1% on Ego4D. Once again, we believe that this indicates overfitting to EgoPlan-IT. Although the models produce more true negatives than false negatives in the TFIV phase of the SV process, even if the model introspects that it was wrong, it often produces the same answer again. Therefore, these models fail to reason logically about their own outputs.

SC significantly degraded performance, but was also the only setting where we had to parse open-ended generation of answers. On the one hand, the original VideoLLaMA models did better at producing reasoning chains, but failed to include a candidate answer, whereas the models trained with EgoPlan-IT tended to generate just the candidate answers—though not consistently or correctly. Generating and extracting precise, structured information from LLMs remains an open challenge.

[2]Picking one sample from the training set and manually writing the reasoning [23].

[3]The original authors did not include this variant in their results [15].

TABLE II: EgoPlan-Bench scores of VideoLLaMA models, some with fine-tuning on EgoPlan-IT, some with fine-tuning on additional datasets, and with additional prompt methods. The chance rate is 25%. † VideoLLaMA with no additional training. ‡ We fixed an error in image processing of the Ego4D benchmark, resulting in slightly improved scores over the original work [15].

# Params	Fine-tuned ViT	Fine-tuning Datasets				Evaluation							
		LLaVA-Instruct + CC-SBU + InternVideo	EgoPlan-IT	RoboVQA	PhysObjects	EPIC-KITCHENS Prompt Method				Ego4D‡ Prompt Method			
						Base	CoT	SV	SC	Base	CoT	SV	SC
7B†	✖	✔	✖	✖	✖	0.278	0.282	0.280	0.007	0.300	0.306	0.300	0.004
13B†	✖	✔	✖	✖	✖	0.262	0.259	0.262	0.004	0.302	0.302	0.302	0.008
7B	✖	✔	✔	✖	✖	0.543	0.523	0.439	0.310	0.445	0.438	0.393	0.286
7B	✖	✔	✔	✔	✖	0.547	0.550	0.488	0.321	0.430	0.430	0.421	0.318
7B	✖	✔	✔	✖	✔	0.543	0.542	0.495	0.306	0.436	0.434	0.427	0.299
7B	✖	✔	✔	✔	✔	0.523	0.526	0.457	0.313	0.433	0.428	0.421	0.328
7B	✔	✔	✔	✖	✖	0.587	0.583	0.439	0.361	0.448	0.442	0.442	0.320
7B	✔	✔	✔	✔	✔	0.559	0.547	0.507	0.307	0.438	0.433	0.440	0.318

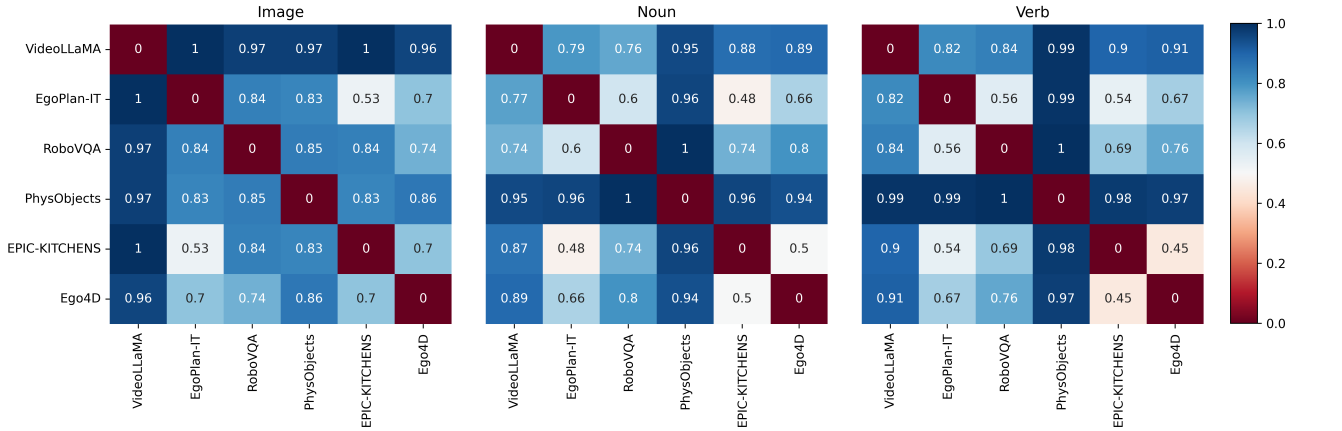

Fig. 3: Wasserstein distances (min-max normalized) between features (based on images, nouns, and verbs) of each dataset, calculated using 1000 random samples per dataset for the images, and the entire vocabularies for the nouns and verbs. The "VideoLLaMA" field refers to the combination of the LLaVA-Instruct-150K, CC-SBU, and InternVideo datasets.

C. Dataset analysis

Given the fine-tuning results, we performed an analysis into the (dis)similarities between the different datasets used, based on both the vision and language modalities (Figure 3). To compare the visual features of each dataset, we randomly sampled 1000 images/keyframes per dataset, extracted vision embeddings using the pre-trained CLIP ViT-B/32 model [42], and calculated pairwise Euclidean distances. We then used the pairwise distances to calculate the Wasserstein-1 distance (Appendix VI-A) between all datasets, using the GenSim library [43], [44]. For ease of comparison, we min-max normalized the distances in Figure 3. We performed the same process with text features for each dataset. Firstly, we extracted the nouns and verbs using the "WordNet" lemmatizer from the NLTK library [45], and then extracted word embeddings using the "fasttext-wiki-news-subwords-300" model [46], [47] via the GenSim library. As before, we calculated the Wasserstein-1 distances between all datasets using Euclidean pairwise distances.

When comparing image embeddings, the "VideoLLaMA" dataset (LLaVA-Instruct + CC-SBU + InternVideo) is the most dissimilar to all other datasets. While it contains images and videos from diverse sources, the other datasets all emphasize an egocentric perspective. As EgoPlan-IT is derived from the EPIC-KITCHENS dataset, which the in-domain evaluation set is also based on, their similarity is to be expected. EgoPlan-IT is also closer to Ego4D than RoboVQA and PhysObjects.

When comparing word embeddings, the simplistic nature of PhysObjects' QA-format stands out. Compared to both RoboVQA and PhysObjects, EgoPlan-IT language domain is still the closest to that of both evaluation datasets.

In summary, the lack of improvement from co-fine-tuning with additional datasets appears to be tied to the dissimilarities in the visual features and text prompts, and a smaller percentage of EgoPlan-IT in the training data distribution.

VI. DISCUSSION

Our experiments and dataset analysis indicate that performance on EgoPlan-Bench is largely dictated by the proportion of EgoPlan-IT data, as including apparently related datasets fails to improve either in-domain or out-of-domain performance. Despite their use in other domains, commonly used prompting methods were not beneficial for EgoPlan-Bench. The models seemed unable to reason properly about their outputs—a finding echoed in other research [40], [48]. However, our results are limited to relatively small foundation models, and may not apply so clearly to larger or future VLMs.

APPENDIX

A. Wasserstein distance

The Wasserstein distance is a measure of the similarity between two distributions. Letting $X = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{d \times n}$ and $Y = (y_1, y_2, \ldots, y_m) \in \mathbb{R}^{d \times m}$ be d-dimensional vectors, the Wasserstein-p distance is formally defined as:

$$W_p(a,b) = \left(\min_{\gamma \in \mathbb{R}^{n \times m}} \sum_{i,j} \gamma_{ij} \ ||x_i - y_j||_p \right)^{\frac{1}{p}} \quad (1)$$
$$\text{s. t.} \quad \gamma \mathbf{1_m} = \mathbf{a}; \gamma^{\mathbf{T}} \mathbf{1_n} = \mathbf{b}; \gamma \geq \mathbf{0}$$

where $\mathbf{1_c}$ is an identity matrix of dimension c. For the datasets comparisons, we used the Wasserstein-1 metric, also referred to as the Earth Mover's Distance (EMD) and commonly used to measure the distance between distributions over words in natural language[49], [44], [50].

REFERENCES

[1] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al.*, "On the Opportunities and Risks of Foundation Models," *arXiv preprint arXiv:2108.07258*, 2021.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language Models are Few-Shot Learners," in *NeurIPS*, 2020.

[3] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–35, 2023.

[4] K. Kawaharazuka, T. Matsushima, A. Gambardella, J. Guo, C. Paxton, and A. Zeng, "Real-World Robot Applications of Foundation Models: A Review," *arXiv preprint arXiv:2402.05741*, 2024.

[5] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, *et al.*, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," in *CoRL*, 2023.

[6] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao, "Look Before you Leap: Unveiling the Power of GPT-4V in Robotic Vision-language Planning," *arXiv preprint arXiv:2311.17842*, 2023.

[7] T. Kwon, N. Di Palo, and E. Johns, "Language Models as Zero-Shot Trajectory Generators," in *Workshop on Language and Robot Learning: Language as Grounding, CoRL*, 2023.

[8] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," in *CoRL*, 2023.

[9] P.-L. Guhur, S. Chen, R. G. Pinel, M. Tapaswi, I. Laptev, and C. Schmid, "Instruction-Driven History-Aware Policies for Robotic Manipulations," in *CoRL*, 2023.

[10] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, *et al.*, "A Generalist Agent," *TMLR*, 2022.

[11] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation," in *CoRL*, 2023.

[12] OpenAI, "GPT-4V(ision) System Card," OpenAI, Tech. Rep., 2013.

[13] L. Chen, M. Zaharia, and J. Zou, "How is ChatGPT's Behavior Changing Over Time?" *arXiv preprint arXiv:2307.09009*, 2023.

[14] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, S. Zhong, B. Yin, and X. Hu, "Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond," *ACM Trans. Knowl. Discov. Data*, 2023.

[15] Y. Chen, Y. Ge, Y. Ge, M. Ding, B. Li, R. Wang, R. Xu, Y. Shan, and X. Liu, "EgoPlan-Bench: Benchmarking Egocentric Embodied Planning with Multimodal Large Language Models," *arXiv preprint arXiv:2312.06722*, 2023.

[16] H. Zhang, X. Li, and L. Bing, "Video-LLaMA: An Instruction-Tuned Audio-Visual Language Model for Video Understanding," *arXiv preprint arXiv:2306.02858*, 2023.

[17] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, *et al.*, "Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100," *IJCV*, pp. 1–23, 2022.

[18] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, *et al.*, "Ego4D: Around the World in 3,000 Hours of Egocentric Video," in *CVPR*, 2022.

[19] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically Grounded Vision-Language Models for Robotic Manipulation," *arXiv preprint arXiv:2309.02561*, 2023.

[20] P. Sermanet, T. Ding, J. Zhao, F. Xia, D. Dwibedi, K. Gopalakrishnan, C. Chan, G. Dulac-Arnold, S. Maddineni, N. J. Joshi, *et al.*, "RoboVQA: Multimodal Long-Horizon Reasoning for Robotics," *arXiv preprint arXiv:2311.00899*, 2023.

[21] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," in *NeurIPS*, 2022.

[22] Y. Weng, M. Zhu, F. Xia, B. Li, S. He, S. Liu, B. Sun, K. Liu, and J. Zhao, "Large Language Models are Better Reasoners with Self-Verification," in *EMNLP*, 2023.

[23] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," *arXiv preprint arXiv:2203.11171*, 2022.

[24] 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," 03 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[25] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-Training with Frozen Image Encoders and Large Language Models," in *ICML*, 2023.

[26] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra, "ImageBind: One Embedding Space to Bind Them All," in *CVPR*, 2023.

[27] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual Instruction Tuning," in *NeurIPS*, 2024.

[28] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models," *arXiv preprint arXiv:2304.10592*, 2023.

[29] J. Chen, D. Zhu, X. Shen, X. Li, Z. Liu, P. Zhang, R. Krishnamoorthi, V. Chandra, Y. Xiong, and M. Elhoseiny, "MiniGPT-v2: Large Language Model As a Unified Interface for Vision-Language Multi-Task Learning," *arXiv preprint arXiv:2310.09478*, 2023.

[30] 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.

[31] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "GPT-4 Technical Report," *arXiv preprint arXiv:2303.08774*, 2023.

[32] 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 *ECCV*, 2014.

[33] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training Language Models To Follow Instructions With Human Feedback," in *NeurIPS*, 2022.

[34] 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.

[35] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in Time: A Joint Video and Image Encoder for End to End Retrieval," in *ICCV*, 2021.

[36] K. Q. Lin, J. Wang, M. Soldan, M. Wray, R. Yan, E. Z. Xu, D. Gao, R.-C. Tu, W. Zhao, W. Kong, *et al.*, "Egocentric Video-Language Pretraining," in *NeurIPS*, 2022.

[37] C. Zhu, F. Xiao, A. Alvarado, Y. Babaei, J. Hu, H. El-Mohri, S. Culatana, R. Sumbaly, and Z. Yan, "EgoObjects: A Large-Scale Egocentric Dataset for Fine-Grained Object Understanding," in *ICCV*, 2023.

[38] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint arXiv:2106.09685*, 2021.

[39] E. Saravia, "Prompt Engineering Guide," 12 2022. [Online]. Available: https://github.com/dair-ai/Prompt-Engineering-Guide
[40] K. Valmeekam, M. Marquez, and S. Kambhampati, "Can Large Language Models Really Improve by Self-Critiquing Their Own Plans?" *arXiv preprint arXiv:2310.08118*, 2023.
[41] R. Hong, H. Zhang, X. Pang, D. Yu, and C. Zhang, "A Closer Look at the Self-Verification Abilities of Large Language Models in Logical Reasoning," *arXiv preprint arXiv:2311.07954*, 2023.
[42] 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*, 2021.
[43] R. Rehurek and P. Sojka, "Gensim–Python Framework for Vector Space Modelling," *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, p. 2, 2011.
[44] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, *et al.*, "Pot: Python optimal transport," *JMLR*, vol. 22, no. 78, pp. 1–8, 2021.
[45] S. Bird, "NLTK: The Natural Language Toolkit," in *COLING/ACL Interactive Presentation Sessions*, 2006.
[46] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *NeurIPS*, 2013.
[47] T. Mikolov, É. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in Pre-Training Distributed Word Representations," in *LREC*, 2018.
[48] K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati, "On the Planning Abilities of Large Language Models - A Critical Investigation," in *NeurIPS*, 2023.
[49] L. V. Kantorovich, "Mathematical Methods of Organizing and Planning Production," *Manag. Sci.*, vol. 6, pp. 366–422, 1960.
[50] S. Otao and M. Yamada, "A Linear Time Approximation of Wasserstein Distance With Word Embedding Selection," in *EMNLP*, 2023.