OPTIMIZING GPT FOR VIDEO UNDERSTANDING: ZERO-SHOT PERFORMANCE AND PROMPT ENGINEERING

Mark Beliaev, Victor Yang, Madhura Raju, Jiachen Sun, Xinghai Hu
Tiktok Inc.

ABSTRACT

In this study, we tackle industry challenges in video content classification by exploring and optimizing GPT-based models for zero-shot classification across seven critical categories of video quality. We contribute a novel approach to improving GPT's performance through prompt optimization and policy refinement, demonstrating that simplifying complex policies significantly reduces false negatives. Additionally, we introduce a new decomposition-aggregation-based prompt engineering technique, which outperforms traditional single-prompt methods. These experiments, conducted on real industry problems, show that thoughtful prompt design can substantially enhance GPT's performance without additional finetuning, offering an effective and scalable solution for improving video classification systems across various domains in industry.

1 INTRODUCTION

In the past decade, recognition over multi-modalities has become an increasingly critical challenge for large video platform companies, such as TikTok and YouTube, which rely on managing vast amounts of user-generated content. Unlike traditional classification tasks that process single-modality inputs, multimodal classification combines multiple data sources, such as image, audio, and text, to provide more accurate and context-aware predictions. The ability to reason across these different modalities is essential for effective content classification. This capability is particularly important for identifying inappropriate content, or videos that do not align with platform policies, as it ensures a more nuanced understanding of content beyond just visual or textual cues.

With the advent of large language models (LLMs) like GPT-4, a new paradigm has emerged in the field of multimodal classification. LLMs offer the promise of generalization, boasting the ability to perform a wide range of tasks with minimal task-specific data, often through one-shot or few-shot learning. Their large-scale pretraining on diverse datasets imbues them with a form of "world knowledge" that can be flexibly adapted to a range of problems, including those that involve multimodal inputs. As a result, generative AI models are now being explored as a promising solution for problems traditionally addressed by multimodal architectures. One domain ripe for this exploration is video classification, where the challenge lies in understanding video content through a combination of visual, auditory, and sometimes textual cues.

However, while LLMs like GPT-4 offer broad generalization, applying them to real-world, industry-specific problems has proven difficult. These models are typically pretrained on massive public datasets, meaning that company-specific data—especially multimodal data related to proprietary video content—is rarely seen during pretraining. As such, LLMs may struggle to capture the nuances of these specific tasks without further tuning. Additionally, despite the hype surrounding generative models, there is still skepticism in the industry regarding their return on investment.

In this study, we aim to address these challenges by evaluating the performance of GPT-4 in real-world, industry-relevant video classification tasks that require multimodal understanding. Specifically, we explore the potential of `GPT-4o` for classifying TikTok video content based on a variety of criteria. Our work differs from traditional multimodal models in two key ways: first, by focusing on the zero-shot and few-shot capabilities of GPT-4, we investigate whether large, pretrained generative models can be effectively applied to multimodal tasks without extensive fine-tuning. Second, we explore the impact of policy refinement and prompt engineering on GPT-4's performance, addressing

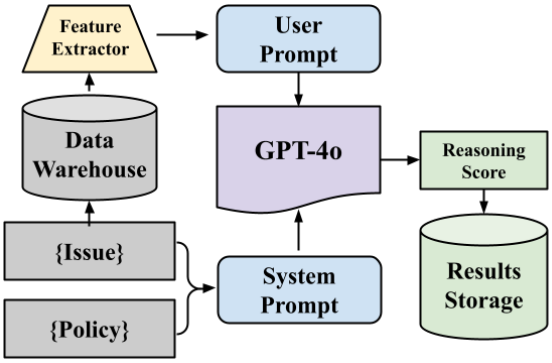

Figure 1: To have a fair comparison across categories, we design the experiment such that the item specific user prompt is independent of the category's policy, while the system prompt provided to `GPT-4o` incorporates the provided policy. For our experiments, given an {CATEGORY} and {POLICY} along with a corresponding dataset, containing at least the *item_id* and ground truth *label*, we ask `GPT-4o` to output a classification prediction, providing its *reasoning* and *score*.

one of the core limitations of LLMs in industry—namely, their difficulty in handling complex, highly specific policies like those used to moderate video content.

Our contributions are twofold. First, we present a comprehensive evaluation of GPT-4's ability to handle real-world, multimodal video classification tasks, comparing its performance to existing, in-house classification models that leverage multimodal features. We show that GPT-4 can perform on par with these models in several categories, but faces challenges in more complex classifications, such as clickbait videos. Second, we introduce novel techniques to enhance GPT-4's performance in these tasks. We find that shortening complex policy prompts improves GPT-4's ability to classify videos accurately by reducing false negatives, and that prompt engineering—specifically, dividing tasks like clickbait detection into subcategories—can lead to significant performance gains. Another highlight is our experiments are grounded in actual industry datasets and problems, making the results highly applicable to video classification tasks at scale.

2 Related Work

Multi-modal Classification. Historically, multimodal modeling has been approached using modality-specific architectures. For instance, BERT (Devlin, 2018), RoBERTa (Liu, 2019), ResNet (He et al., 2016), and ViT (Dosovitskiy, 2020) are commonly employed for text and vision tasks, respectively. Multi-modal models like CLIP (Radford et al., 2021) and GLIP (Li et al., 2022) have effectively combined visual and textual inputs, leveraging correlations between modalities to achieve strong classification and detection results. However, these models typically require extensive task-specific fine-tuning and are limited by the rigid nature of their underlying architectures. This has resulted in a fragmented landscape where models are highly specialized for narrow tasks and lack generalizability to unseen problems or broader domains without significant retraining.

Large Generative Models. In addition to classic tasks like recognition and detection, generative models have become a critical component of modern multi-modal learning, given that they are adaptable to different tasks. Recent advancements include models like Flamingo (Alayrac et al., 2022), OpenFlamingo (Awadalla et al., 2023), LLaVA (Liu et al., 2024), InstructBLIP (Dai et al., 2023), and GPT-4o (Achiam et al., 2023), which integrate multiple modalities for content generation and reasoning tasks. Of particular interest for this study is GPT-4o, the state-of-the-art at the time of this project, which we utilize to evaluate content safety issues on video platforms. These generative models, unlike their classification counterparts, show greater adaptability and can be applied to a broader range of tasks with less task-specific tuning.

3 PROBLEM AND METHOD

This section details our methodology for testing the capability of GPT-4o at identifying various Feed Quality (FQ) categories across TikTok. We first provide some relevant background to the problem at hand in Section 3.1, following which we formulate how we test GPT-4o as a vision-language classifier in Section 3.2. We proceed by detailing our results and analysis in Section 4, and conclude by discussing some limitations and future directions in Section 5.

3.1 PROBLEM SETUP

We define the problem of identifying several domain specific categories across TikTok abstractly, as a binary classification task: given a sample TikTok video post, our goal is to identify whether the content is an example of the category at hand. We refer to posts which correspond to the category at hand as *positive* cases, using the binary label 1, while considering all other posts as *negative* cases, using the binary label 0. TikTok has For You feed eligibility standards that promote an appropriate experience for broad audiences by limiting sexually suggestive content, shocking & graphic content, content that tricks or manipulates others as a way to increase engagement, and unoriginal content. Each FQ category is defined by clear policy and guidelines that support consistent labeling for classification tasks. The resulting human annotations serve multiple purposes, including evaluating and monitoring algorithms aimed at identifying these categories.

Two major challenges to this problem are: (1) Training high quality annotators is difficult and time consuming. (2) Using solely human annotation for monitoring is impractical to support millions of daily posts. To this end, it is desirable to have intelligent systems which can perform well at the defined classification task across a plethora of FQ categories, preferably requiring little to no training. Hence with the recent releases of LLMs equipped with the ability to process multimodal inputs, it is natural to question whether we can utilize these off the shelf models for our task. Particularly, we are interested in the performance of GPT-4o at classifying TikTok videos according to the defined text policies, treating it as black-box classifier, with no additional training. Such a result would not only provide us with insight on GPT-4o's vision-language understanding, but also serve as a comparison for application specific models currently employed, as well as a baseline for more costly approaches requiring additional training.

3.2 METHOD

In this section, we detail how we implement GPT-4o as a vision-language classifier for FQ categories in our experiments, and refer the reader to the official GPT-4o API used for reference OpenAI (2025). We provide a high level overview of our experiment design in Figure 1. Given a FQ category defined by the text strings {CATEGORY}, the name of the category, and {POLICY}, the detailed text policy for the corresponding category, we provide GPT-4owith the following *system* level instructions:

Task: Label Videos for {CATEGORY}Content
Objective:
You are required to classify videos based on if they are {CATEGORY}content. The classification will help in training models to identify {CATEGORY}content. Each video should be labeled with an associated score indicating the likelihood of the video being {CATEGORY}content.
Detailed Policy:
{CATEGORY}:
{POLICY}
Output:
For each video, output a clear reasoning behind your decision and a score (0-100) indicating the overall likelihood of the video being {CATEGORY}content.
Format your output as a JSON object with the following keys:
reasoning: a chain of reasoning that explains how you arrived at your classification.
score: An integer score from 0 to 100 representing how likely it is that the video is {CATEGORY}content, with 100 indicating that the video certainly is {CATEGORY}content, and 0 indicating that you are confident this is not {CATEGORY}.

Notes:
- Be clear and specific in your classification.
- Inspect every frame provided.
- Use the detailed policy to guide your judgment.

Following this, for each TikTok video post we extract a set of text features: (audio transcription, hashtags, text, sticker text), alongside the video frames as base64 encoded images. Given this set of features, we query `GPT-4o` with the following *user* level prompt, providing the video frames alongside:

Given a video with the following features:
    audio transcription: {asr}
    hashtag: {hashtag}
    text: {text}
    sticker text: {stickerText}
    video frames: images with base64 encoding provided
Format your output as a JSON object with the specified keys.

Based on our specified output format, we collect a JSON object containing the specified keys for each video, the reasoning used by `GPT-4o` as a text string, and the prediction score as an integer ranging from 0 to 100.

4 RESULTS

4.1 TASKS

We conducted experiments using OpenAI's latest `GPT-4o` model, accessed through Microsoft Azure at the time of the study. The temperature parameter was set to zero, with the top $p = 1$ value fixed at one, without employing any stop-words. Additionally, the frequency penalty was set to 0.5, while the presence penalty remained at zero. The Gemini version we compare in our preliminary study is Gemini-1.5-pro. For each video, we sampled frames at 0.5 frames per second, including the first and last frames, and used a maximum of 30 frames irrespective of the video length.

Using the formulation defined in the previous section, we test the performance of `GPT-4o` across 7 FQ categories at TikTok:

- **Sensitive and Mature Themes**: Nudity and body exposure or sexually suggestive content
- **Shocking and Graphic Content**: Content that may intentionally shock, upset, or disgust others.
- **Non-Interactive Modules**: More than 2 content modules appear in one video, but there is no real communication, connection, or mutual response between them.

- **Clickbait**: A tactic that tricks users to interact with TikTok video or accounts through follow, like, share, comment, finish and other actions in order to artificially get more traffic than they honestly would.
- **Static Frame (SF)**: Video format but completely static, including pictures, solid color, screenshots.
- **Watermark**: The video includes watermarks of other social media platforms and apps.
- **Usefulness**: Content that conveys knowledge, experience, information that helps users to learn more.

For each category, we use a balanced dataset of at least 500 video posts, and collect the reasoning and score provided by `GPT-4o`. Each video post has an associated ground truth label provided by a human annotator, and when available, a *baseline* score, which corresponds to the normalized output of the current production model being used to identify this category at TikTok. Note that while the individual categories typically represent a small portion of the overall TikTok feed, in order to have a fair comparison across categories, we use balanced datasets containing equal number of positive and negative cases.

4.2 BASELINES

For the domain specific tasks we called out earlier, we build classification models that leverages multi-modal features and human annotation results for training as the baselines. The major difference between the baseline models and `GPT-4o` is baselines models are either rule-based or trained with annotated samples, while the `GPT-4o` is pretrained on large vision-language corpora with a larger model size. Specifically, for SF/Non-Interactive/Watermark, we could not collect enough training data to train a classifier by the time of this experiment. We tried rule-based method on these issues and only got acceptable performance on SF. We omit reporting the baseline performance for Non-Interactive/Watermark in our experiment.

4.3 EXPERIMENTAL SETUP

Given these datasets, we perform three experiments which we overview below:

Exp. 1 We test the performance of `GPT-4o` across the 7 FQ categories defined, displaying the results in Figure 2 as precision-recall curves for each category, and providing additional statistics as well as a comparison to the baseline production models in Table 1. For all categories, we compare the AUC (Area Under the Curve) score, along with the total portion of samples that are false positives and false negatives, for both `GPT-4o` and the respective baseline model. We sort the categories by their AUC, provide the word count of each policy (# words). For Non-Interactive/Watermark, the baselines are weak and thus we don't present the numbers here.

Exp. 2 We perform an experiment on the Sensitive & Mature category to test the affect of shortening the policy provided in the system prompt, displaying our results in Figure 3.Specifically, we summarize the policy of Sensitive & Mature category from 4023 words to XXX words, while comparing it with the Non-Interactive category at the same time, where the policy word number are just 116.

Exp. 3 We perform an experiment on the Clickbait category to test if we can further improve `GPT-4o` one-shot performance with simple prompting engineering, displaying our results in Figure 4. Specifically, we split the clickbait into 8 subcategories, ask `GPT-4o` to make independent prediction on these subcategories and then aggregate the scores into one combined prediction for this category.

4.4 ANALYSIS

We analyze our research by answer the following 6 questions sequentially, utilizing the aforementioned experiments.

Q1. How does `GPT-4o` perform at identifying FQ categories? Using the results from **Exp. 1**, we take the score output by `GPT-4o`for each category, and plot the corresponding precision-recall curves in Figure 2 by sweeping through all possible classification threshold values. We see the best performing categories are Non-interactive and Shocking & Graphic Content, while all other

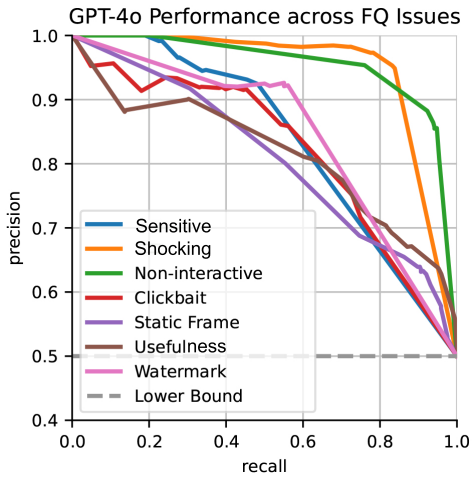

Figure 2: This figure shows the precision-recall curves when using the score provided by `GPT-4o`for all categories. Since all datasets are balanced to contain equal numbers of pos & neg cases, they share one lower bound (0.5).

Table 1: `GPT-4o` vs. Baseline Models

Task		`GPT-4o`			Baseline		
category	# words	AUC	FP	FN	AUC	FP	FN
Non-Interactive	116	0.94	0.08	0.03	-	-	-
Shocking	934	0.91	0.01	0.17	0.97	0.01	0.12
Usefulness	143	0.83	0.18	0.08	0.90	0.03	0.22
Clickbait	713	0.79	0.02	0.30	0.89	0.02	0.26
Static Frame	80	0.79	0.26	0.04	0.79	0.11	0.13
Watermark	402	0.76	0.02	0.23	-	-	-
Sensitive	4023	0.73	0.00	0.41	0.95	0.01	0.25

categories seem to be rather close in performance. This demonstrates a gap between certain categories which are simple to classify for `GPT-4o`, and categories which are more difficult. We explore this gap further in **Exp. 2** which is analyzed in **Q3-5** below.

Q2. How does the one-shot performance of `GPT-4o` compare with current production models? We can see from our results of **Exp. 1** displayed in Table 1 that `GPT-4o` performs on par with the production model baselines for Static Frame, Non-interactive, and Watermark. For Clickbait, Shocking & Graphic Content, and Useful, the performance of `GPT-4o` is slightly lower (≤ 0.1 AUC). For SD/SF/Watermark, since the baselines are rule-based due to lack of training data, `GPT-4o` shows the advantage on leveraging world knowledge and reasoning from generative pretraining. Finally, we note that `GPT-4o` shows the worst relative and absolute performance on Sensitive & Mature. We continue our analysis by discussing what factors can contribute to the gaps in `GPT-4o`'s performance between categories.

Q3. What are the characteristics of categories that are difficult for `GPT-4o`? Looking closer at our results from **Exp. 1** by analyzing Table 1, we can make the following observations: 1) All baselines suffer performance due to False Negatives (FN) instead of False Positives (FP). 2) `GPT-4o`'s predictions for Non-interactive, Useful, and Static suffer more from FP errors compared to FN errors. 3) `GPT-4o`'s predictions for Clickbait, Watermark, Sensitive, and Shocking suffer more from FN errors compared to FP errors. The first observation is expected as production models are typically designed to be conservative, minimizing the affect that incorrect predictions have on the overall user experience. For the second and third point, we highlight that this pattern corresponds directly to the number of words in the policy: categories defined with less complex policies (which are easier to express in natural language), are dominated by FP errors instead of FN errors.

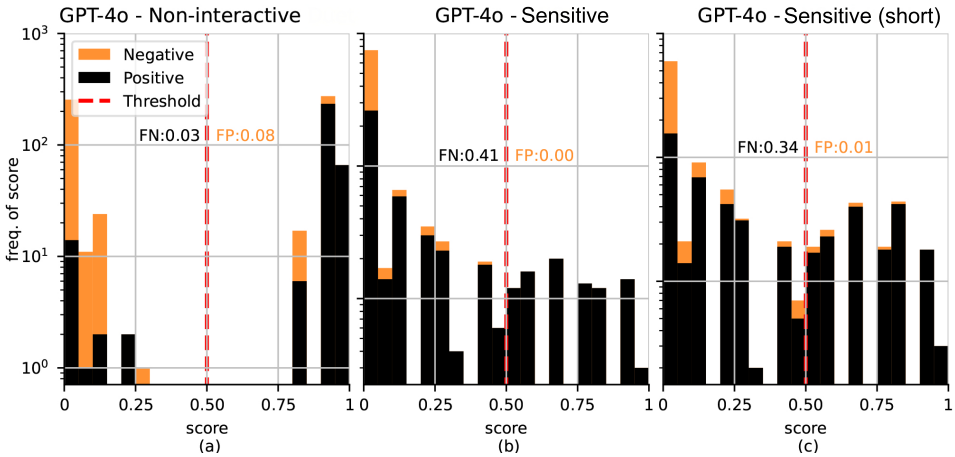

Figure 3: These charts plot the normalized (0-1) distribution of scores (x-axis) provided by GPT-4o, where the dashed red line is the classification threshold we choose, orange and black are used to depict negative and positive cases, and a logarithmic scale is used on the y-axis. We show the results for Non-interactive duet in (a), Sensitive and Mature Content in (b), and a shortened version of Sensitive and Mature Content in (c), which uses 96 words in the policy instead of 4023. We also report the total portion of False Negatives (FN) in the dataset: which are true positives (black), but classified as negative (left of the threshold), and conversely, the total portion of False Positives (FP), which are true negatives (orange), but classified as positive (right of the threshold).

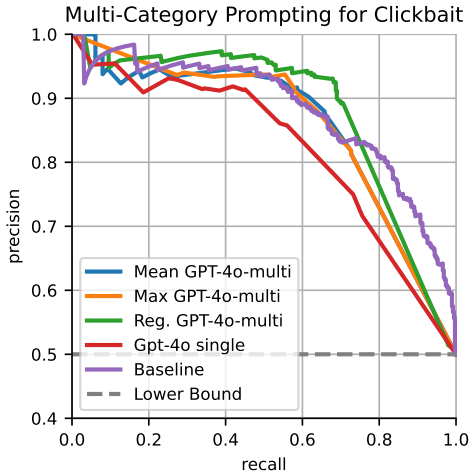

Figure 4: This figure shows the precision-recall curves for the Clickbait category when using both the original prompting technique (GPT-4o-single), as well as the proposed prompting technique which asks to provide a score for each category (GPT-4o-multi). To produce a final score for GPT-4o-multi, we consider the mean, max, as well as the best linear regression of the category scores. We additionally plot the corresponding production model (Baseline).

To further explore this last point, we can look at our results for **Exp. 2** displayed in Figure 3. Looking first at the left plot (a), we see that the score output by GPT-4o for the Non-interactive category follows the desired trend: there is a valley in the middle, positive cases have high scores, and negative cases have low scores. The reported FP and FN are 0.08 and 0.03 respectively. Looking at the middle plot (b), we see the performance on Sensitive & Mature is much worse: although FP is 0.00, FN is 0.40. The plot clearly shows that this is not just a trade off induced by the threshold choice, as there are many more positive (black) videos to the left of the threshold. Finally, looking at the right plot (c), we see the performance of GPT-4o when we shorten the Sensitive & Mature policy from 4023 words to only 96: the reported FN decreased by 0.07 points, while FP increased by only 0.01. The

new AUC (not shown) using the short prompt is 0.79, which is significantly larger than the previous 0.73.

Q4. Why does the performance of GPT-4o on Sensitive & Mature improve when we shorten the policy? We believe our results support the following hypotheses: when the policy is long and specific, GPT-4o is conservative - outputting a low score when it can not find correlations between the policy and the features. Since the policy is very detailed, GPT-4o paints with a fine brush, and is unlikely to mislabel negative videos as they do not correlate with the specific instructions in the policy (contributing to low FP). However, since many edge cases need to be considered for categories that require detailed policies, it is hard to find all positive cases (contributing to high FN). Conversely, when the policy is short, GPT-4o is aggressive - since a shorter policy has to make generalizations, GPT-4o paints with a broader brush and can identify more positive cases (contributing to low FN). However, negative videos are likely to be mislabeled due to this (contributing to high FP).

Q5. How can we extrapolate this result beyond Sensitive & Mature Issues? We conclude our discussion of **Exp 2.** by stating that these results do not provide a full story. We have to recall that shortening the prompt did not make most of the errors generated by FP cases, it only lowered the portion of FN cases. This is expected, we can identify more positive videos because we are lowering the specificity of what we consider as Sensitive & Mature content. However, removing information from the prompt did not suddenly make the this category easier, like the Non-interactive Duet category. Although one may observe from our results that categories with large prompt size are dominated by FN errors, we can not say that this is due to prompt length alone. However, categories that inherently require a specific policy because they are hard to express in natural language due to the many edge cases present, can be characteristic of such behavior. Furthermore, long and detailed prompts are not solely responsible for poor performance, as observed by looking at the results for Shocking & Graphic in Table 1.

In conclusion, if we want good performance at identifying positive cases, we should consider optimizing the prompt length provided to GPT-4o, trying to express the category with as few words as possible, while still covering all relevant cases. Before concluding our work, we discuss some more ways to improve GPT-4o's one-shot performance.

Q6. What steps can we take to improve GPT-4o's one-shot performance?

Throughout our study, we have tried several common methods for improving the one-shot performance of GPT-4o, such as manual prompt refinement and few-shot in-context instruction learning. Neither of these techniques provided significant performance improvement, and hence are not discussed here. However, we found that for the Clickbait category, which has a policy of 713 words splitting the category into 8 subcategories, asking GPT-4o to provide a score for each category gave favorable results. Specifically, we consecutively prompt GPT-4o to provide a reasoning and score for the corresponding category of Clickbait, and combine these 8 scores by either taking the mean, the max, and the linear regression. We can see the results of **Exp. 3** in Figure 4, showcasing that all three techniques (GPT-4o-multi) produced better results than the previous method (GPT-4o-single). Interestingly, we found that performing linear regression on the scores allowed us to outperform the production model for higher precision values (which is the typical, more conservative, regime of interest.) Overall, we see that application specific prompt-engineering can lead to substantial performance improvement when using GPT-4o, without any additional training.

5 Conclusion

In this paper, we investigate the application of GPT-4o on video classification and provide a strong baseline on the application of LLMs in video classification and other multimodal tasks. By demonstrating the potential of generative models in solving practical industry problems, we offer insights that can guide the development of more effective, scalable solutions for industries that rely on multimodal data. Moreover, our findings challenge the notion that LLMs are not suited for industry-specific applications, showing that with the right adaptations, these models can be both powerful and practical tools for real-world classification problems.

Limitations: We do not consider finetuning, as our focus is one shot performance. Additionally, we do not list results of other VLM models (e.g. Gemini), but found that initial experiments showed very similar performance between them.

REFERENCES

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

Dai, W., Li, J., Li, D., Tiong, A., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S. Instructblip: Towards general-purpose vision-language models with instruction tuning. arxiv 2023. *arXiv preprint arXiv:2305.06500*, 2, 2023.

Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Li, L. H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10965–10975, 2022.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Liu, Y. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

OpenAI. Openai official api documentation, 2025. URL `https://openai.com/index/openai-api/`.

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