Master Thesis Proposal

Enhancing Relational Understanding in CLIP Leveraging Hard Negative Captions (HNC)

Author: Wen Wen
st186079@stud.uni-stuttgart.de

Supervisors: Pascal Tilli
Maksym Sevkovych

Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart,
Germany
pascal.tilli@ims.uni-stuttgart.de

Tech & AI Lab
Seedbox Ventures
Marienstraße 27, 70178 Stuttgart,
Germany
maksym.sevkovych@seedbox-ai.com

January 16, 2025

1 Introduction

In the rapidly developing field of artificial intelligence, Vision Language Models (VLMs) have become a critical area of research, enabling machines to simultaneously understand both visual and textual information (Lu et al., 2019; Li et al., 2019; Radford et al., 2021; Li et al., 2021, 2022; Singh et al., 2022). These models underpin various applications, such as Visual Question Answering (VQA), content moderation, and robotics. However, while significant progress has been made with models like CLIP (Contrastive Language-Image Pretraining) (Radford et al., 2021), critical challenges persist in relational reasoning, particularly in distinguishing nuanced object relationships within scenes.

CLIP's vision encoder has demonstrated superior performance in object recognition and zero-shot tasks by aligning image and text embeddings through large-scale contrastive pre-training. Despite these strengths, it struggles with relational reasoning tasks. For instance, distinguishing between '*The bus is on the road*' and '*The road is on the bus*' remains a significant challenge. CLIP achieves a semantic similarity of only 53.39% for the incorrect match and 46.61% for the correct one(Huang et al., 2023). This highlights a fundamental weakness in CLIP's ability to effectively model relationships. The inability to encode relationships effectively impacts downstream tasks such as VQA, especially in zero-shot scenarios where the model must generalize to unseen combinations of visual and textual inputs. This limitation arises because the training of CLIP primarily focuses on object and text associations, often neglecting relational semantics.

One promising approach to address this limitation is leveraging negative samples (Yuksekgonul et al., 2023; Huang et al., 2023). They point out that Vision-Language Models tend to behave like bags-of-words. As a result, training solely on these datasets has proven insufficient in addressing the limitations in compositional understanding. Therefore, negative samples are introduced where the word order is intentionally scrambled, further aiding the model's ability to discern compositional relationships. For example, in the previous scenario, telling CLIP '*The bus is on the road*' is correct while telling it '*The road is on the bus*' is incorrect. Additionally, the Hard Negative Captions (HNC) dataset (Dönmez et al., 2023) provides another valuable resource to improve relationship understanding in vision language models. This dataset provides negative samples with more subtle differences, making it harder for the model to distinguish between correct and incorrect relationships. This fine-grained difficulty forces the model to learn more subtle distinctions between objects and their relationships, thereby enhancing its ability to understand complex scenarios.

My research focuses on enhancing the relationship understanding capabilities of vision encoder, with a specific focus on CLIP. I propose to address this limitation by leveraging Hard Negative Captions that explicitly highlight relational distinctions during fine-tuning the vision encoder. By exposing CLIP to examples that contrast incorrect and correct object relationships, the model can be trained to better differentiate between them. These minimally contradictory samples force the model to develop a deeper understanding of relationships while maintaining strong object recognition performance. Improving relational understanding in vision encoder is an impactful research area that addresses critical limitations in existing models. This work contributes to building robust multimodal systems capable of reasoning with greater precision and relational awareness, advancing applications in VQA and beyond.

2 Related Work

Vision-Language Models (VLMs) VLMs bridge visual and textual modalities, enabling tasks such as VQA, image-text retrieval, image captioning, and visual grounding. Prominent models typically employ transformer-based architectures and large-scale multimodal pre-training to learn joint embeddings of images and text. ViLBERT(Lu et al., 2019) uses a two-stream BERT architecture, with separate encoders for vision and language, and incorporates co-attention mechanisms to align and integrate visual and textual features, making related features closer in the shared embedding space. Similarly, LXMERT(Tan and Bansal, 2019) adopts a two-stream design, where image and text features are processed separately before being combined using cross-attention layers, with a cross-modality encoder that aligns and integrates information between the two modalities. In contrast, UNITER(Li et al., 2019) employs a single-stream transformer that processes both image and text features simultaneously, allowing for more highly integrated cross-modal interactions. CLIP(Radford et al., 2021), on the other hand, employs contrastive learning to align images and text within a shared embedding space, pulling positive matching pairs closer together while pushing negative matching ones apart. This approach has made CLIP the most widely used model, owing to its versatility and impressive zero-shot capabilities. While CLIP excels at recognizing objects and linking them to textual descriptions, it faces challenges in understanding complex relationships between objects in a scene, such as spatial or semantic interactions.

NegCLIP Several approaches have been proposed to address these limitations in CLIP's ability to understand object relationships. One such method is NegCLIP (Yuksekgonul et al., 2023). It proposes the core issue identified is that many VLMs, including CLIP, often behave like "bags of words" — they struggle with relational understanding and can perform well on tasks like image-text retrieval without understanding the complex relationships between objects. NegCLIP tackles this problem by introducing composition-aware hard negative mining during training. This approach generates negative samples by perturbing the composition or order of words in captions and pairing them with similar images. This forces the model to distinguish between correct and incorrect orderings, enhancing its ability to learn relational and compositional structures. However, this approach also has limitations. For instance, when the word order changes in a sentence, such as "*Black and white cows*" being swapped to " *White and black cows*," NegCLIP might incorrectly label the new sentence as a negative sample, even though the meaning remains the same. This issue arises because the model treats word order as a critical factor, even when it may not change the intended meaning. Such low-quality negative samples will limit the performance of the model.

Structure-CLIP Structure-CLIP (Huang et al., 2023) enhances NegCLIP by incorporating Scene Graph Knowledge (SGK) to address the limitations of random word swapping in generating negative samples. It ensures that generated negatives focus on meaningful semantic alterations, such as object-attribute or object-relationship pairs, while avoiding cases like swapping non-critical words such as 'Black' and 'White' in "*Black and white cows*," as such changes do not affect the phrase's meaning. However, the key word exchange approach can be problematic when the verb represents a bidirectional relationship. For example, swapping "*A man walks with a dog*" to "*A dog walks with a man*" results in a grammatically and semantically correct sentence that should not be treated as a negative sample. This highlights that while Structure-CLIP improves upon NegCLIP's random sampling, it may still struggle with nuanced contexts where word-order changes do not alter the underlying semantics. Addressing these cases may require more sophisticated methods for generating negative samples.

HNC Hard Negative Captions (HNC)(Dönmez et al., 2023) generates negative samples that are minimally contradictory to their corresponding images, making them more challenging for models to discern compared to typical unrelated image-text mismatches. HNC leverages scene graph information (Krishna et al., 2016) to systematically create negative captions by representing objects, attributes, and relationships within an image. The process involves creating positive captions that accurately reflect parts of an image and then generating hard negatives by making minimal changes to the positive captions to create semantically close but mismatched descriptions. These changes include attribute-based alterations, such as modifying an object's attributes (e.g., "The bowl is white" → "The bowl is black"), relation-based changes like altering relationships between objects (e.g., "The cat is on the table" → "The cat is under the table"), and reasoning-based adjustments involving logical structures like "AND" or "XOR" relations. The negative samples in HNC are intentionally closer to the positive captions to increase the model's difficulty in distinguishing between them. For example, for an image showing a white cat on a red chair, the positive caption could be "*The white cat is sitting on the red chair.*" while a hard negative caption might be "*The white cat is sitting on the orange chair.*" By introducing these fine-grained, compositionally complex negative samples, the HNC method and its dataset enable models to better differentiate subtle mismatches and improve their ability to reason about intricate visual and linguistic alignments.

3 Goals

The primary goal of this research is to enhance the relational understanding capabilities of CLIP while preserving its strong object recognition performance. This will be achieved by fine-tuning CLIP using the Hard Negative Captions (HNC) dataset, which provides positive samples paired with minimally contradictory negative samples.

The fine-tuning process will involve training the model to minimize the similarity between an image and its corresponding positive caption while maximizing the similarity between the image and its hard negative caption and other random negative captions. Based on the traditional CLIP loss function, the hard negative loss will be added with weight. By using negative samples that closely resemble the positive ones, the model will be forced to capture finer-grained features and develop a more nuanced understanding of object relationships.

The key objective is to create a fine-tuned version of CLIP that excels in both object recognition and relational reasoning, overcoming the limitations of the original model. The resulting model will be a robust vision-language system capable of handling complex scenarios requiring relational awareness, with potential applications in tasks like Visual Question Answering (VQA) and other multimodal reasoning tasks.

4 Material and Methods

4.1 Data Preparation

4.1.1 Training Datasets

The training process incorporates the following datasets:

- GQA Dataset[1] : The GQA (Graph Question Answering) dataset provides

[1] GQA Dataset Link

images and corresponding textual descriptions, focusing on visual reasoning tasks.

- HNC Dataset[2] : Based on GQA, the HNC (Hard Negative Captions) dataset is generated to introduce more challenging textual negatives for contrastive learning.

4.1.2 Sample Selection and Pairing Method

For each image I_i, there are multiple positive text descriptions T_{ij} [3]. Different from CLIP, hard negative captions are introduced here. The corresponding hard negative captions T_{ij}^{HNC} are specially designed negative samples that incorporate challenging perturbations of the positive texts for relational reasoning. To facilitate training, these relationships are stored as triplets: $(I_i, T_{ij}, T_{ij}^{HNC})$, $j \in \{1, 2, \ldots, M_i\}$, where M_i is the total number of positive-HNC pairs for image I_i.

During batch selection, a specific number of triplets are randomly sampled from the entire set of stored triplets to form a batch. To avoid complicated situdation, only one positive-HNC pair for image I_i will be reserved. For convenience, the subsequent triplets are written as: (I_i, T_i, T_i^{HNC}). In such a case, the pairs for image I_i in each batch include following parts: (I_i, T_i) is the positive pair indicating the text description that correctly matches the image. (I_i, T_i^{HNC}) is the hard negative pair, which serves as the primary source for enhancing the model's relational reasoning capabilities, helping it better distinguish between subtle relational differences. Following the original CLIP formulation, cross-image negative pairs are created by using texts T_j and T_j^{HNC} from other images I_j $(j \neq i)$ as random negative samples for the current image I_i, so the negative pairs are (I_i,T_j) and (I_i,T_j^{HNC}), as illustrated in Figure1.

To facilitate computation, the pairs can be divided into two square matrices: the positive matrix, same as the traditional CLIP, and hard negative matrix, an additional part introduced by this proposal. In the positive matrix, the diagonal elements represent the positive pairs, while the off-diagonal elements represent random negative pairs. In the hard negative matrix, the diagonal elements represent the hard negative pairs, while the off-diagonal elements also represent as random negative pairs, seen as in Figure2.

[2] HNC Dataset Link

[3] Here in T_{ij}, j denotes the positive index for that image.

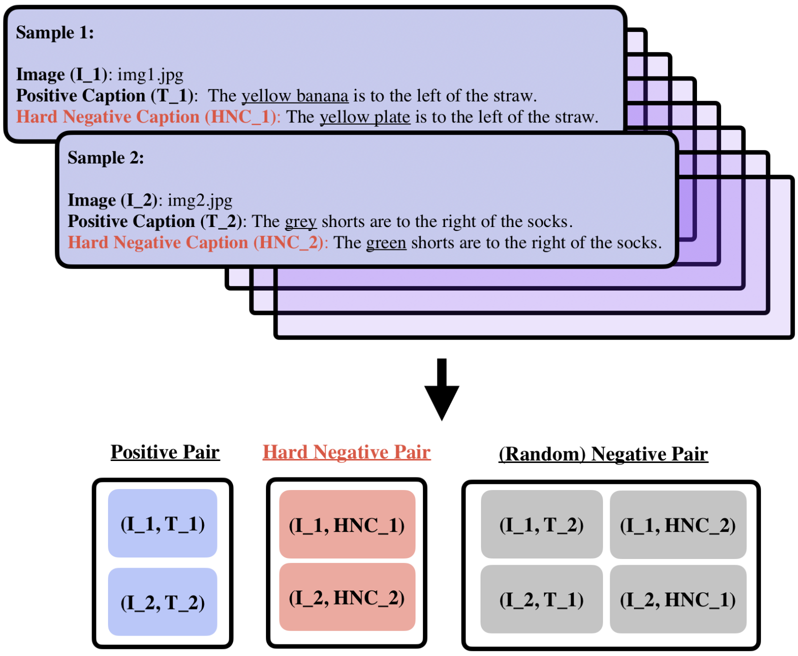
Sample 1:
Image (I_1): img1.jpg
Positive Caption (T_1): The yellow banana is to the left of the straw.
Hard Negative Caption (HNC_1): The yellow plate is to the left of the straw.
Sample 2:
Image (I_2): img2.jpg
Positive Caption (T_2): The grey shorts are to the right of the socks.
Hard Negative Caption (HNC_2): The green shorts are to the right of the socks.
Positive Pair
Hard Negative Pair
(Random) Negative Pair
(I_1, T_1)
(I_2, T_2)
(I_1, HNC_1)
(I_2, HNC_2)
(I_1, T_2)
(I_1, HNC_2)
(I_2, T_1)
(I_2, HNC_1)

Figure 1: Illustration of the Data Pairing. Here take two samples as an example.

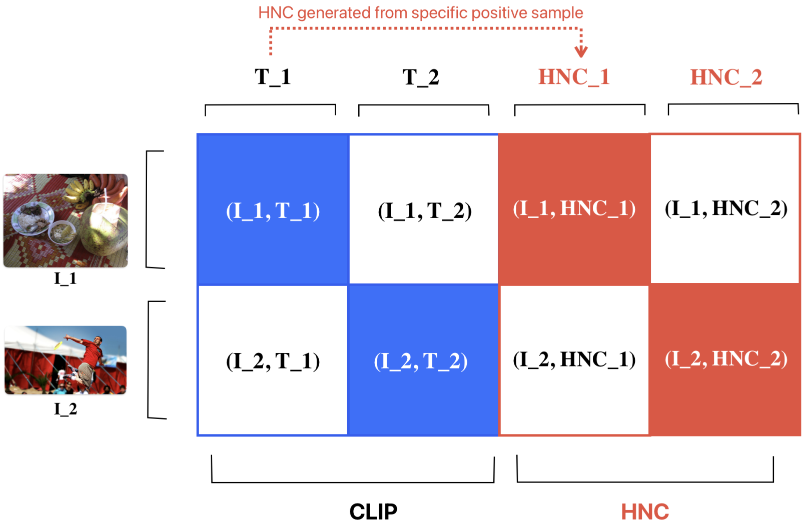

Figure 2: Pairing Matrix. The blue areas represent positive pairs, the red areas indicate hard negative pairs, and the white areas correspond to random negative pairs.

4.2 Loss Function

The contrastive loss function in CLIP comprises two parts: loss for the image alignment ($Loss_i$) and loss for text alignment ($Loss_t$). Based on the original CLIP loss function, an additional Hard Negative Contrastive loss is introduced. A weight parameter w_i is used to control the contribution of the HNC loss, as showed in Figure3.

Loss Function of image to text

$$\mathcal{L}_i = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(\frac{sim(I_i,T_i)}{\tau}\right)}{\exp\left(\frac{sim(I_i,T_i)}{\tau}\right) + \sum_{j\neq i} \exp\left(\frac{sim(I_i,T_j)}{\tau}\right) + \boxed{w_i} \exp\left(\frac{sim(I_i,T_i^{HNC})}{\tau}\right) + \sum_{j\neq i} \exp\left(\frac{sim(I_i,T_j^{HNC})}{\tau}\right)}$$

+

Loss Function of text to image

$$\mathcal{L}_t = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(\frac{sim(T_i,I_i)}{\tau}\right)}{\exp\left(\frac{sim(T_i,I_i)}{\tau}\right) + \sum_{j\neq i} \exp\left(\frac{sim(T_i,I_j)}{\tau}\right) + \boxed{w_i} \exp\left(\frac{sim(T_i,I_i^{HNC})}{\tau}\right) + \sum_{j\neq i} \exp\left(\frac{sim(T_i,I_j^{HNC})}{\tau}\right)}$$

Figure 3: Loss Function of Image-to-Text and Text-to-Image

Here $Sim(i,t) = \frac{i \cdot t}{\|i\| \cdot \|t\|}$ represents the cosine similarity between embeddings.

To maintain the original capabilities of the CLIP model, an L2 regularization term is added as a penalty term to prevent significant changes in the model parameters. The total loss is defined as:

$$\mathcal{L} = \frac{1}{2}(\mathcal{L}_i + \mathcal{L}_t) + \lambda \sum_i (\theta_i - \theta_i^{CLIP})^2.$$

- θ_i is the current model parameter.
- θ_i^{CLIP} is the original pretrained CLIP parameter.

4.2.1 Hyper Parameters in Loss Function

Here are some hyper parameters in loss function can be adjust to compare and gain a better performance.

- w_i controls the influence of the hard negative sample. The larger w_i, the stronger emphasis on the hard negative sample, encouraging the model to focus more on distinguishing between the image and the specifically challenging hard negative text. Smaller w_i values can be utilized to mitigate the impact of hard negatives when their influence significantly diminishes the model's overall recognition performance.

- τ is the temperature parameter which adjusts the scale of similarity scores. A smaller τ accentuates differences in similarity scores, enhancing the model's sensitivity to distinctions between positive and negative samples. In contrast, a larger τ reduces these differences, promoting smoother optimization and diminishing the influence of individual outliers.
- λ controls the strength of the L2 regularization. A larger λ strongly enforces the preservation of the original CLIP parameters, limiting changes during fine-tuning but potentially hindering the model's adaptability to the new task. Conversely, a smaller λ provides the model with greater flexibility to adapt, albeit with a higher risk of forgetting the original CLIP capabilities.

4.3 Training Procedure

The training process includes the following steps:

1. Loading and Pairing Data:
   - Match positive text and its corresponding HNC to each image $(I_i, T_{ij}, T_{ij}^{HNC})$.
   - For each batch, randomly sample M triplets, ensuring that only one triplet is selected per image.
2. Loading Model and Reference Model:
   - Load vision encoder and text encoder of CLIP. Freeze the text encoder, only upgrade parameters in vision part.
   - Load another vision encoder of CLIP as a reference model, for computing the changing of parameters.
3. Forward Pass:
   - Using the CLIP vision encoder and text encoder to transfer images and texts into tensor.
   - In each batch, separate pairs into: $(I_i, T_i)$and (I_i, T_i^{HNC}) for compute convince.
   - Calculate similarity scores.
4. Loss Calculation

- Compute the total loss based on the previous loss function.

5. Backpropagation and Optimization:

   - Perform backpropagation to calculate gradients of $\mathcal{L}$ with respect to model parameters.
   - Update the model parameters using an optimizer (e.g., SGD, Adam).
   - Only update the parameters of the vision encoder, keeping the text encoder's parameters frozen. As the text encoder has already learned rich textual representations from large-scale text datasets. Fine-tuning it can lead to over fitting or degrade its language understanding capabilities. By keeping the text encoder's parameters frozen, we leverage its robust textual feature extraction capabilities while focusing on adapting the vision encoder to better align with the textual representations. This approach is computationally more efficient as it reduces the number of parameters to update, leading to faster convergence and lower memory usage.

4.4 Evaluation

The performance of the fine-tuned model will be evaluated in downstream tasks by integrating it into the LLaVA (Large Language and Vision Assistant)(Liu et al., 2024) framework. Specifically, we evaluate the model on the widely-used visual question answering (VQA) task, which tests multimodal reasoning capabilities. The evaluation focuses on two datasets:VQA v2.0 and TextVQA, which together provide a comprehensive benchmark for general VQA performance and relational reasoning. The performance of the fine-tuned model is compared against CLIP, serving as the baseline vision encoder in the LLaVA architecture.

4.4.1 Integration with LLaVA Framework

The LLaVA framework enables seamless multimodal reasoning by combining a vision encoder with a large language model (LLM) via a multimodal connector. In our evaluation:

- We replace the vision encoder in LLaVA with our fine-tuned model while keeping the LLM and connector unchanged.

- The modified LLaVA architecture processes image-question pairs and generates natural language answers, leveraging both visual and textual context.

4.4.2 Evaluation Datasets

VQA v2.0[4]: General VQA Benchmark VQA v2.0 is a widely-used dataset for open-ended visual question answering tasks. It features a diverse range of questions that test the model's ability to understand objects, attributes, and spatial relationships within images. Example questions include:

- *"What color is the car in the foreground?"*
- *"How many people are standing near the tree?"*

The dataset includes a balanced distribution of yes/no, number, and open-ended questions, making it a robust benchmark for evaluating general-purpose VQA models.

TextVQA[5]: Relational and Text-Rich Benchmark TextVQA focuses on questions that require understanding and reasoning about text within the visual context. This dataset is particularly challenging as it requires the model to recognize text in images (e.g.signs, labels, license plates) and understand relationships between text and objects, such as positions or associations. Example questions include:

- *"What is written on the billboard above the car?"*
- *"Which direction does the sign point to?"*

TextVQA provides a benchmark for assessing the model's ability to integrate textual elements with visual context, emphasizing relational reasoning.

4.4.3 Evaluation Metrics

To evaluate the performance of the model on VQA v2.0 and TextVQA, we adopt standard evaluation metrics consistent with widely-used practices in the field.

Accuracy Accuracy is the primary metric used to measure the performance of the model. It evaluates how many questions are answered correctly relative to the ground truth answers.

- For **VQA v2.0**, accuracy is calculated using partial credit based on annotator agreement:

$$Accuracy_{VQA} = \min\left(\frac{Votesfortheanswer}{3}, 1\right)$$

[4] VQA v2.0 Dataset Link

[5] TextVQA Dataset Link

where "Votes for the answer" is the number of annotators (out of 10) who agreed on the answer. This formula gives full credit for unanimous agreement and partial credit for lower levels of agreement.

- For **TextVQA**, accuracy is calculated as an exact match between the model's generated answer and the ground truth. Full credit is given for exact matches, and no partial credit is awarded.

$$Accuracy_{TextVQA} = \frac{CorrectAnswers}{TotalQuestions}$$

Comparison with Baseline Models To highlight the improvements achieved by our model, we compare its performance against the baseline CLIP-based LLaVA model. Metrics are reported for both datasets and across different question types where applicable.

5 Time Plan

The following table outlines the timeline for this project, highlighting key milestones for each month:

Month	**Milestone**
February	Writing literature review and preparing the datasets
March	Initial implementation and debugging for fine-tuning components
April	Completing fine-tuning and use more different hyper parameters to generate more fine-tuned models for further evaluation.
May	Model evaluation on VQA v2.0 and TextVQA datasets
June	Finishing the comparison of evaluations and writing the main results section.
July	Finalizing the thesis paper and preparing for submission

Table 1: Project Timeline with Monthly Milestones

References

Esra Dönmez, Pascal Tilli, Hsiu-Yu Yang, Ngoc Thang Vu, and Carina Silberer. 2023. HNC: Leveraging hard negative captions towards models with fine-grained visual-linguistic comprehension capabilities. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 364–388, Singapore. Association for Computational Linguistics.

Yufeng Huang, Jiji Tang, Zhuo Chen, Rongsheng Zhang, Xinfeng Zhang, Weijie Chen, Zeng Zhao, Zhou Zhao, Tangjie Lv, Zhipeng Hu, and Wen Zhang. 2023. Structure-clip: Towards scene graph knowledge to enhance multi-modal structured representations.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. 2019. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. Flava: A foundational language and vision alignment model.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2023. When and why vision-language models behave like bags-of-words, and what to do about it?