5 Methods

5.1 Dataset Description

For this work, datasets that are used are CCNL1, CCNL2 and CLEVR. The datasets CCNL1 and CCNL2 were created in the Computational Cognition and Neuroscience Lab. at the University of Padova, as part of the ongoing efforts to create reliable benchmarks using realistic datasets to evaluate the counting ability of AI models. The Compositional Language and Elementary Visual Reasoning (CLEVR) is a synthetically generated dataset that is composed of visual scenes with 3D shapes. CLEVR was originally created as a tool to allow researchers to test visual reasoning capabilities of AI models [methods-1]. As one can see, the images of CLEVR do not include just one type object shape: each scenery includes geometrical shapes of diverse colour and shapes. In Fig 5.1 sample images from CLEVR for each numerosity are displayed. All images across the datasets used in this study are annotated with the number of distinct objects present, covering numerosities from 1 to 10.

The CCNL1 dataset is a synthetic dataset and consisting of 2-dimensional objects with variable sizes displayed on a uniform background [9]. The object categories butterflies, apples, people, dots and fastcards. There are also 3 additional object categories (triangles, circles and squares) with black background. For each combination of numerosity and object category, the dataset includes 50 samples, with object positions randomized within each image. An illustration of the CCNL1 dataset is shown in Figure 5.2.

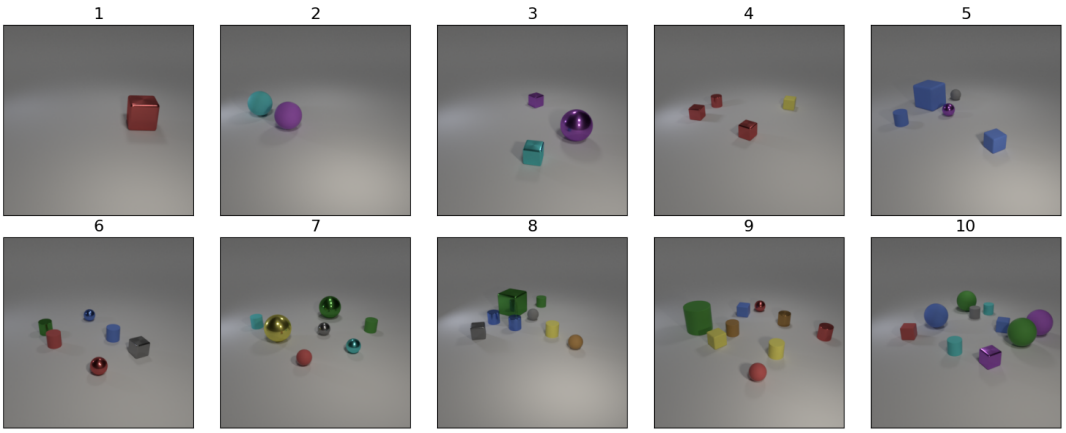


Figure 5.1: The Samples from the CLEVR Dataset: Each image contains objects of different shapes

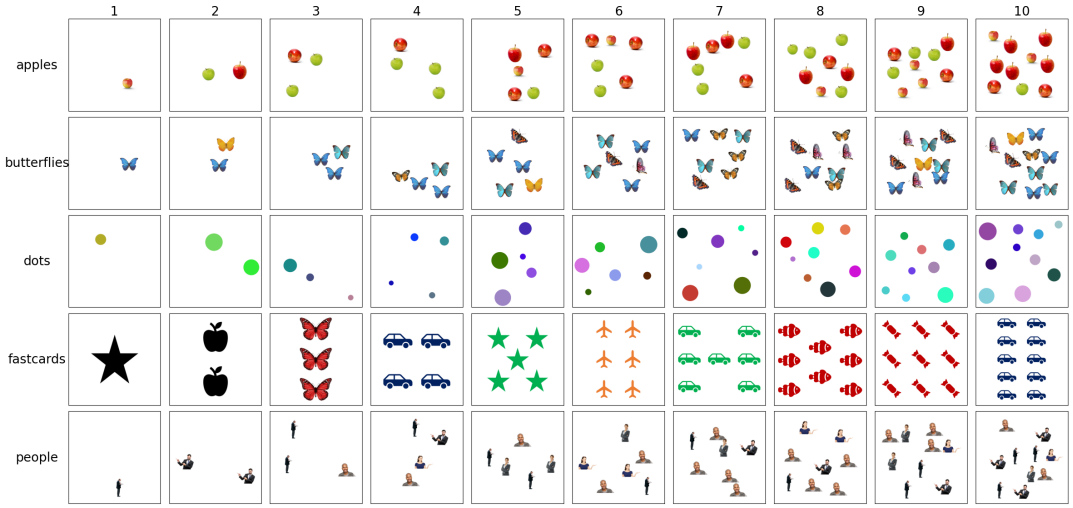


Figure 5.2: The CCNL1 Dataset

Figure 5.3: The CCNL2 Dataset

The CCNL2 dataset consists of images of real-world objects. The images were taken in naturalistic conditions and the lighting conditions, background and object placements vary across the scenes. The object categories that are included are apples, bottles, butterflies, cups and knives. The original images (2992x2992px) were also downsized to (360x360px) to make working with the dataset less computationally demanding. Figure 5.2 illustrates some samples from the CCNL2 dataset.

Since this work focuses on the transfer of visual enumeration across different object categories, datasets, and numerosities, I will use several dataset splits. These splits are summarized in Table 5.1.

Table 5.1: Description of the datasets and splits used in this work.

Dataset / Split	Obj. Categories	Numerosities	Total Samples
CCNL1	Apples, Butterflies, People, Fastcards, Dots	1–10	2500
CCNL1PromptTrain	Circles, Squares, Triangles	1–10	1500
CCNL2	Apples, Bottles, Cups, Knives, Buttons	1–10	2500
CLEVR	Mixed	1–10	500
CCNL2Train	Apples, Cups	1–10	1000
CCNL2Test	Buttons, Bottles, Knives	1–10	1500
CCNL2TrainEven	Apples, Cups	2,4,6,8,10	500
CCNL2TrainOdd	Apples, Cups	1,3,5,7,9	500

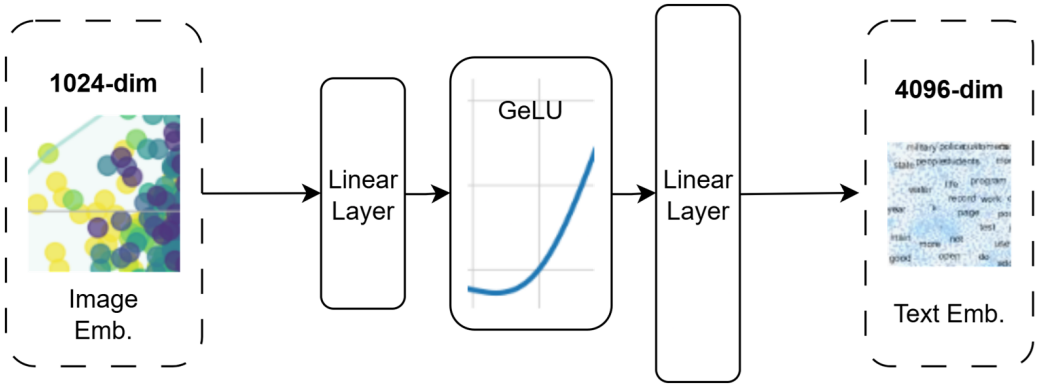


Figure 5.4: The Multimodal Projector

5.2 Finetuning the Multimodal Projector

The multimodal projector of LLaVA consists of 2 linear layers with GeLU activation function between them [8]. With around 20 million parameters, the multimodal projector is the most lightweight component of LLaVA. As a first attempt at improving the visual enumeration performance, I targeted the multimodal projector for QLorA tuning. The architecture of the multimodal projector can be seen in Figure 5.4.

The performance gains were notable when I tested the finetuned model on a test set with similar distribution, but the transfer to other object categories and datasets was modest (for the detailed results please see the Results and Discussion section). Considering that pretraining has been shown to help MMLLMs to retain their performance under distribution shifts, I followed this intuition to see if a lightweight tuning step on the vision-encoder could offer similar improvements.

5.3 Finetuning ViT on Synthetic Data

Building on top of the logic I followed in the last section, finetuning vision transformer may be necessary to obtain projected image embeddings that represent numerosity better. Verma et al. showed that tuning the multimodal projector did not enrich the representations of the projected image embeddings with respect to the task it is tuned for[methods-2]. Working with image embeddings that are better numerosity-aware is the logical next step. As a result, I proceed with supervised and lightweight finetuning of CLIP ViT. Before diving into the process,

it is useful to measure how good are the embeddings of the baseline vision transformer in representing numerosity, explaining the reason why CLIP ViT is not great at producing numerosity-aware embeddings.

5.3.1 How Numerosity Aware Are Image Embeddings of CLIP ViT L/14?

CLIP ViT L/14, the vision encoder of LLaVA 1.5 was pretrained with contrastive learning on a huge image-text dataset collected from the internet (WebImageText). WebImageText contained 400 million pairs of images and their corresponding textual captions. During training, the parameters of the vision encoder and the text encoder were updated so that visual features with their associated textual descriptions are pulled together while non corresponding pairs are pushed apart[37]. The image embeddings of ViT are expected to have excellent semantic separation, making similar concepts cluster together while unrelated ones remain distant in the embedding space. For illustrative purposes, I reduced the dimensionality of the image embeddings of CCNL2Train datasets to 2 using PCA and plotted them in figure 5.5. Each image corresponds to a dot with a hue that varies according to the number of objects in the image.

The plot shows that the embeddings of 'cups' and 'apples' are very well linearly separated. This observation is aligned with the fact that the visual embeddings of the vision transformer have excellent semantic representations: a classifier operating on these embeddings can easily learn to distinguish these 2 categories. Unfortunately, images with different cardinality of object instances are not separated as well. There exists a clear separation between embeddings of images with 1 cup and the images with 10 cups but, the embeddings of the images of 7 cups and 8 cups clearly blend together. This so-called 'blindness' of the visual encoder is not unique to the object counts. Experiments of Rudman et al. demonstrate that the visual encoders of state-of-the-art MMLLMs fail to differentiate between different geometric shapes[methods-4]. The UMAP plot (figure 5.6) reinforces that the separation based on object counts is weak, especially as the number of objects increase.

But why do CLIP ViT struggles with such seemingly simple perceptual tasks? One of the first things to investigate is the data ViT was trained on. OpenAI has not made the dataset that CLIP ViT was trained on public but we can investigate a publicly available one: LAION400. LAION400 is comparable in content to WebImageText[69] The graph below was taken from Testolin et al. and displays the relative frequency of captions within the dataset that include each numerosity [9].

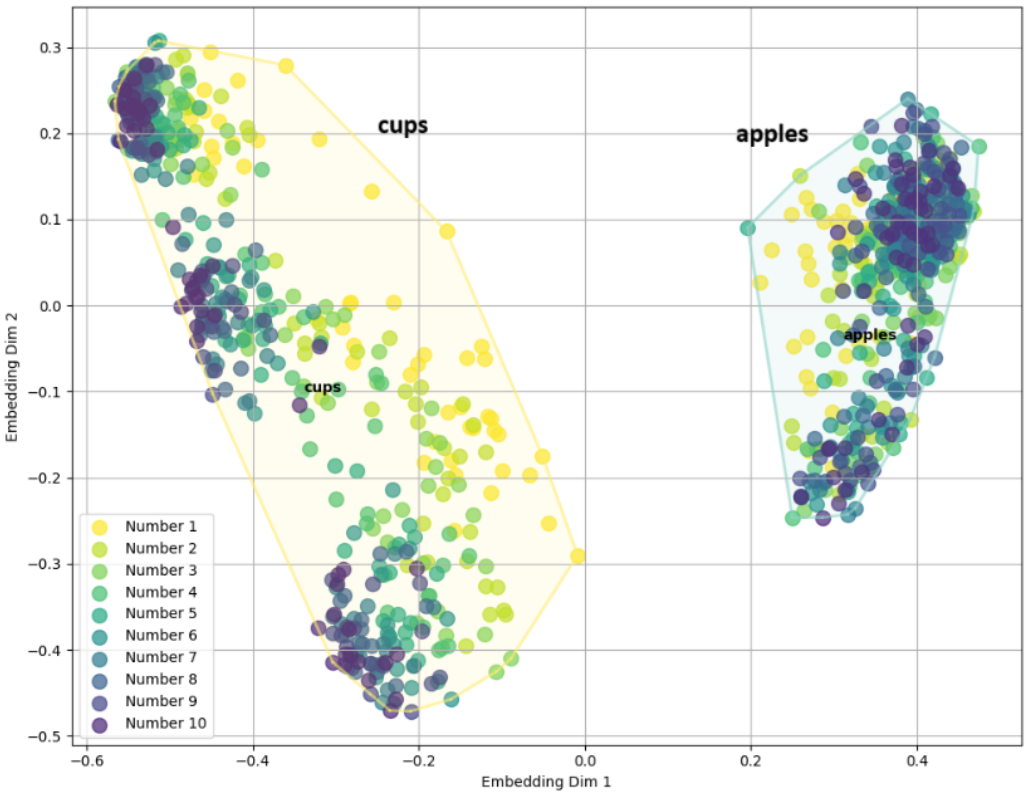


Figure 5.5: ViT Embeddings of CCNL2-Train Dataset. Different colours correspond to different numerosities.

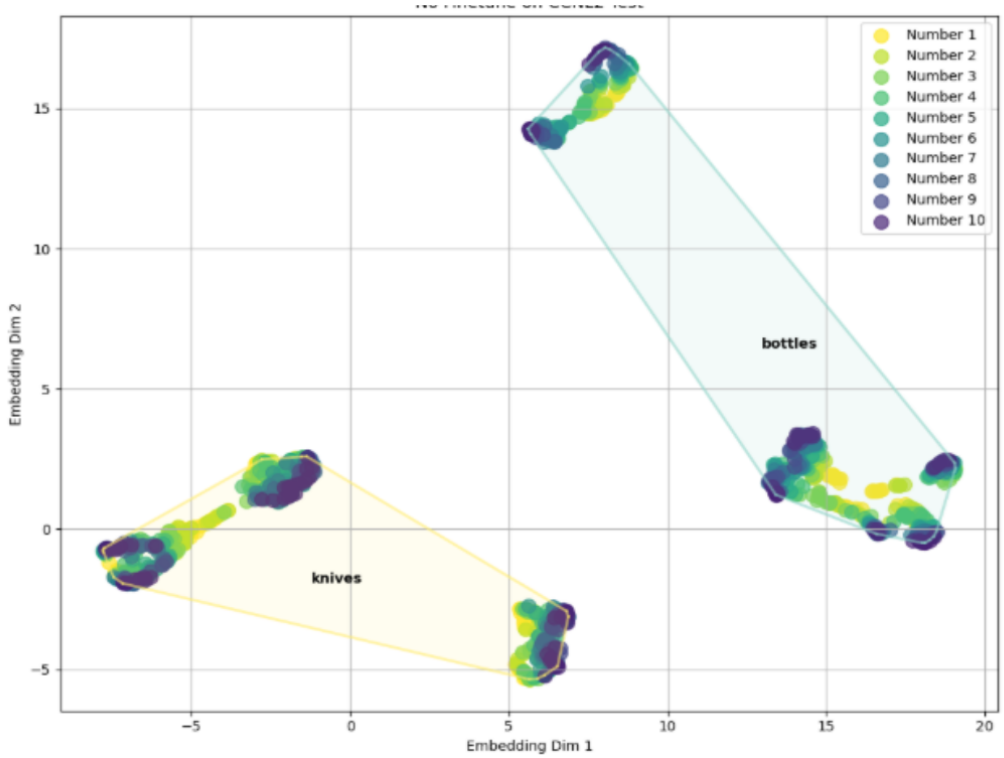


Figure 5.6: UMAP Embedding of CCNL2Train - Baseline ViT. Different colours correspond to different numerosities.

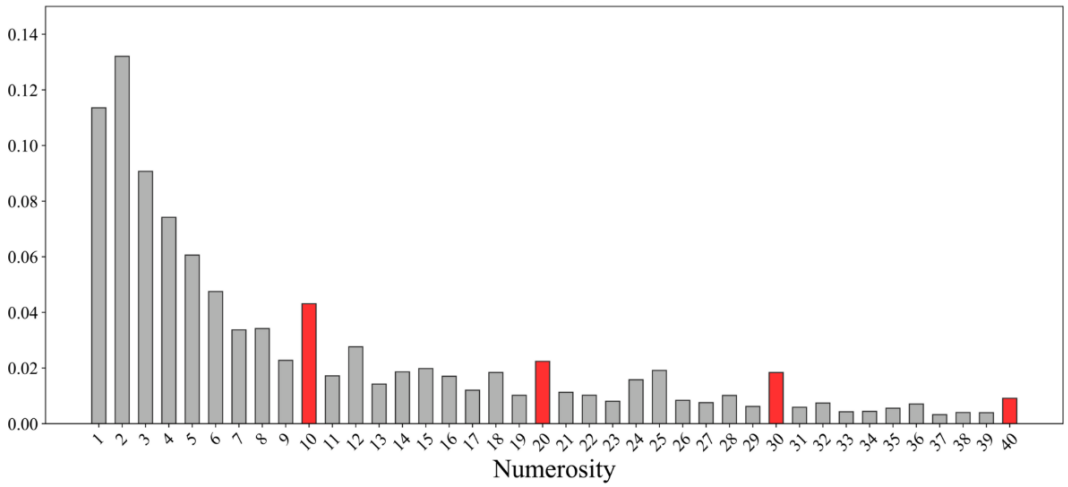


Figure 5.7: Relative Frequency of Each Numerosity in LAION400M: The power-law distribution

The graph shows that the number of objects in the samples can be approximated as a power-law distribution. Such a distribution is not unique to LAION400M and was also observed in Microsoft COCO dataset. As stated by Testolin et al., this uneven distribution of numerosities could be one of the reasons MMLLMs struggle with counting tasks when presented with images having cardinality above a certain value [59][9].

5.3.2 Why Synthetic Data?

I performed all of the finetuning procedures on ViT with CLEVR: an artificially generated dataset containing 3-dimensional shapes in varying shapes, sizes, colours and spatial arrangements. I deliberately chose to work with synthetic data because it offers several advantages over using naturalistic images. According to Liu et al. one of the most prominent upsides of using synthetic datasets for training is the possibility to generate synthetic data in a much greater scale[70]. This is especially true for datasets that require cardinality annotations and other annotations of abstract nature: trying to create diverse naturalistic datasets with cardinality annotations and varying object types is notoriously time consuming. Another notable advantage of using synthetic data is that it enables targeted manipulations that support fine-grained hypothesis testing about the mechanisms underlying numerical perception. This allows the creation of more balanced datasets that are better tailored to specific finetuning scenarios [70]. This can assist researchers a lot in the creation of datasets for numerosity tuning, ensuring the datasets have desired surface-area and contour-length distributions, while allowing researchers to control other factors like amount of occlusion between the objects.

The original CLEVR paper emphasizes the versatility that synthetic dataset creation pipelines offer[71]. The code that was made available by the authors allows researchers to generate visual scenes with customizable properties. The rendering pipeline allows users to control both the properties of the objects in the scene (shape, count, size, etc.) and the relations between the objects (occlusion, front/behind)[1]. Since the release of CLEVR in 2017, numerous variants were proposed including the ones that focus on identifying objects based on referring expressions and the ones that incorporate dynamics into visual scenes[72][72].

5.3.3 Finetuning ViT with PEFT - Considerations

As stated earlier, the main motive of finetuning ViT is to make LLaVA more robust to distribution shifts when testing on visual enumeration. Want et al. stated that pretraining can help

[1] https://github.com/facebookresearch/clevr-dataset-gen/tree/main/image_generation

with robustness to distribution shifts but the advantages of the pretraining can be limited by the bias and lack of diversity in the data that was used for pretraining [73]. While it is hard to guarantee that the CLEVR dataset is fully free of such biases, the wide range of object shapes and diverse object arrangements likely contribute to more robust numerosity representations that in turn will support better downstream performance across different categories and scenarios.

There are a few additional important considerations to take into account when finetuning ViT. One of them is with the pretraining procedure, we have to make sure to not cause catastrophic forgetting. Catastrophic learning is the degradation of performance on previously learned tasks as the AI system learns to perform novel tasks [74]. For our case, ViT should retain most of its semantic understanding that it gained from the contrastive pretraining. The second consideration comes from the hypothesis that numerosity is a high-order latent factor that is highly invariant to low-level details of the visual scenery in an image. Emergence of numerosity as a high-order statistical property was observed in hierarchical deep neural networks that were trained in unsupervised setting [4].

By following the line of thinking discussed above, I have decided to inject QLoRA to the last layers of ViT and frame the task as a supervised classification problem. The classification is done with a linear layer on top of the last layer of ViT. Results that support using LoRA to counter catastrophic forgetting were obtained by Bafghi et al. They showed that DINO ViT-Base/16 suffered from performance degradation on ImageNet-1k after being finetuned on CIFAR100, whereas using LoRA updates for finetuning prevented the catastrophic forgetting, preserving most of the performance on ImageNet-1K [75]. Considering also that the numerosity is likely a high-level statistical feature, I opted to unfreeze only the last few layers ViT. To minimize the possibility of catastrophic forgetting while preserving the expressiveness of the LoRA adapters I only considered LoRA ranks between 8 and 16. The other parameters were selected based on heuristics and a warm-up period was used for training to facilitate more stable training[76]. A high level schema of the training setting is shown in Fig 5.8 while the exact hyperparameters are reported in the Experiments section.

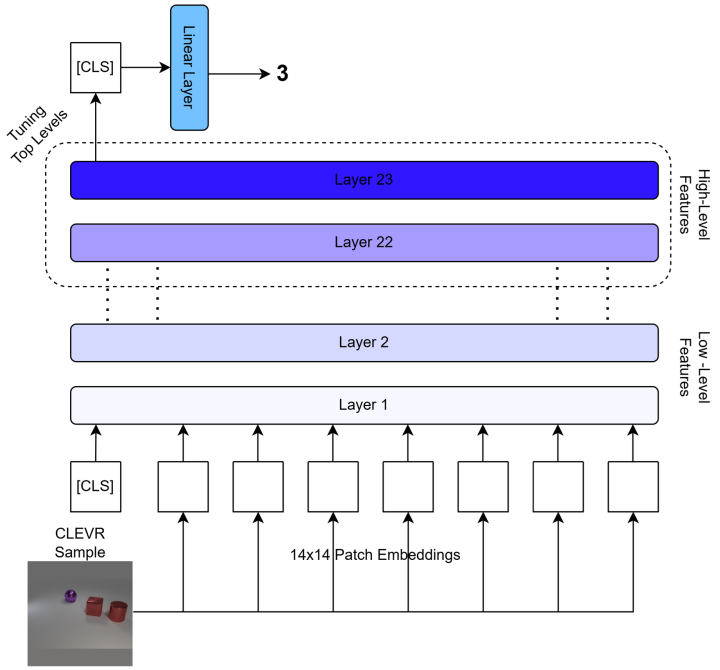


Figure 5.8: The CLIP ViT L/14 Finetuning Setting

5.4 Finetuning of LLaVA for Better Numerosity Again (After Pretraining ViT)

After enhancing the vision transformer for more numerosity-aware image embeddings, I have retrained the entire LLaVA pipeline to observe if the improvements in visual enumeration are more robust to distribution shifts.

5.5 Dimensionality Reduction Techniques

Dimensionality reduction is a crucial pre-processing step in machine learning and data analysis, aiming to reduce the number of variables while preserving essential information. Broadly, these techniques are categorized into linear and non-linear methods.

5.5.1 Linear Dimensionality Reduction Techniques

Linear dimensinality reduction techniques work with assumption that the data can be represented effectively by a lower-dimensional linear subspace. One of the most popular linear dimensionality methods is Principal Component Analysis (PCA) [43] Assuming a dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$ with n samples and d features, PCA seeks a projection matrix $\mathbf{W} \in \mathbb{R}^{d \times k}$ that maximizes the variance of the projected data:

$$\mathbf{Z} = \mathbf{X}\mathbf{W}, \tag{5.1}$$

where $\mathbf{Z} \in \mathbb{R}^{n \times k}$ is the lower-dimensional representation with $k < d$. The projection matrix $\mathbf{W}$ is obtained by solving:

$$\mathbf{W} = \arg\max_{\mathbf{W}} \mathrm{Tr}(\mathbf{W}^\top \mathbf{S}_X \mathbf{W}), \tag{5.2}$$

where $\mathbf{S}_X = \frac{1}{n-1}\mathbf{X}^\top \mathbf{X}$ is the sample covariance matrix, and $\mathrm{Tr}(\cdot)$ denotes the trace. The solution corresponds to the eigenvectors of $\mathbf{S}_X$ associated with the largest eigenvalues. The explained amount of variance is an important quantity that we can check to see if our dimensionality reduction process captured enough variance in the data. The explained amount of variance of the component i can be calculated by:

$$\text{Explained Variance}_i = \frac{\lambda_i}{\sum_{j=1}^{d} \lambda_j} \tag{5.3}$$

where λ_i is the eigenvalue of the ith component and d i the total amount of dimensions. In this case, if we want to calculate how much variance is captured by a 2-dimensional PCA projection we can just sum $\lambda_1 + \lambda_2$ and divide by $\sum_{j=1}^{d} \lambda_j$.

Other notable methods include Linear Discriminant Analysis (LDA) [77] and Factor Analysis (FA)[78].

5.5.2 Non-Linear Dimensionality Reduction Techniques

Non-linear techniques aim to capture intrinsic manifold structures in data that are not possible to fully represent using liner subspaces. Examples include t-SNE [44] and UMAP[45]. The methods are quite good at preserving topology and neighborhood between points but are computationally more intensive and less interpretable than their linear counterparts.

In this work, I will mainly use linear dimensionality reduction techniques. Specifically, I will use PCA as I hypothesize the effects of tuning a linear classifier will be captured better by a linear dimensionality reduction technique.

6 Experiments

In this chapter I will describe the experiments that I conducted. These experiments seek to explain the configuration, hyperparameters and motive of the experiments. LLaVA 1.5-7b-hf[1] checkpoint was the baseline LLaVA model. The training of the LLaVA pipeline was performed on a virtual machine hosted in Google Cloud Compute Engine with an L4 GPU provided by the Computational Cognitive Neuroscience Lab. During all of the experiments mixed-precision training was used to reduce the time requirements. All code for reproducing the experiments is publicly available.[2]

6.1 Testing the Baseline LLaVA Pipeline

I first conducted an experiment to evaluate the performance of the untuned (baseline) LLaVA on the CCNL1 and CCNL2 datasets. This experiment is aimed to identify which categories/numerosities LLaVA find more challenging. In this experiment LLaVA was presented with each image in the CCNL1 and CCNL2 datasets and was prompted with the following prompt:

```
USER: <image>
How many objects are there in the image? Respond only with the number.
ASSISTANT:
```

[1] https://huggingface.co/liuhaotian/llava-v1.5-7b

[2] https://github.com/TimurOner/improving_num_perception_LLaVA

The of the prompt above is a standard for LLaVA 1.5. The '<image>' is a critical token that allows the language model to map the image tokens from the multimodal projector correctly. The inference procedure was done on Google Colab Environment with a NVIDIA L4 GPU. The inference procedure completed in 41 minutes for CCNL2 dataset and in 70 minutes for the CCNL1 dataset. The inference was performed with a batch size of 16.

6.2 Finetuning and Testing Without Partitioning the Numerosities and Object Categories

In the first finetuning experiment, I unfroze only the multimodal projector and used a random 80/20 splitting strategy on the CCNL2Train partition. This splitting strategy resulted similar distributions both in the training and the test set. The point of this experiment is to establish a distribution-matched upper bound for our pipeline: by training and testing on splits drawn from the same distribution (same object categories and numerosity histogram), we measure how well the multimodal projector can perform when no distributional shift is present. The hyperparameters used are shown in Table 6.1.

6.3 Finetuning the Multimodal Projector (Before Finetuning ViT)

My first attempt at improving the counting ability of LLaVA was to tune the multimodal projector. To achieve this I implemented the QLoRA finetuning, a LoRA adapter with rank of 16 was injected into the 4-bit quantized multimodal projector. The rank was selected so that the adapter has enough representation power to improve the model but also to not cause overfitting and catastrophic forgetting. During training only the adapter weights were updated, and all other parameters including the parameters in the language model and vision encoder were kept frozen. The training and validation were performed on CCNL2Train that included the object categories 'apples' and 'cups'. All numerosities from 1 to 10 were included. The hyperparameters used are shown in Table 6.1.

The hyperparameters for training and LoRA adapter fine-tuning were selected based on heuristics from prior work in QLoRA/LoRA research and my preliminary experiments. Due to the high computational cost of training large multimodal models, exhaustive hyperparam-

Table 6.1: Training and LoRA hyperparameters used.

Training Hyperparameter	Value
Learning Rate	1e-5
Weight Decay	1e-4
Batch Size	3
Epochs	8
Optimizer	Adam
LR Scheduler	CosAnnealing

LoRA Hyperparameter	Value
LoRA Rank	16
LoRA Alpha	16
LoRA Dropout	0.05
Target Modules	mmproj.linear[1,2]
Task Type	Causal LM

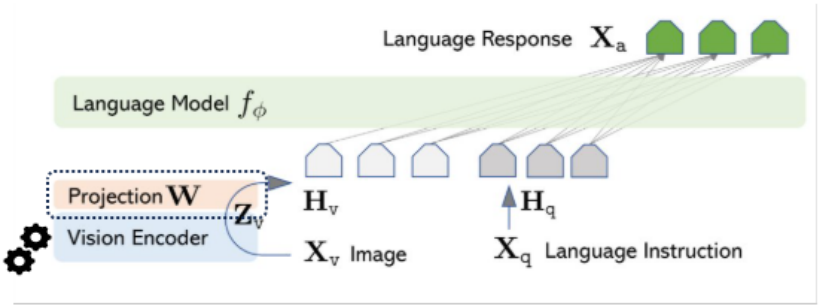


Figure 6.1: Tuning the Multimodal Projector of LLaVA Pipeline: Only the adapters injected in the projector are updated

eter search was not performed. The batch size was constrained with 3 due to computational limitations. The learning rate was kept low to prevent overfitting.

The parameters were quantized using NormalFloat4 (NF4) precision. This helped with preserving the weight distribution. The computations were performed in 16-bit precision to preserve stability during training.

6.4 Finetuning the Visual Transformer (ViT)

In this experiment, the visual encoder of LLaVA is separately tuned to make its representations more numerosity-aware. CLEVR was used for the finetuning process.

For the tuning task, a linear classification setting with a linear layer on top of the CLS token of the vision transformer was considered. QLoRa adapters were injected into the last attention layers of ViT, and all of the parameters of ViT were frozen allowing only the updates of QLoRA adapters.

In addition, I performed dataset augmentation on the CLEVR dataset to make the dataset more rich in terms of different scales, colors, and rotations. The main motive of augmenting

the dataset is forcing ViT to learn more robust number representations that are invariant to rotations, lighting changes, scaling, and other visual variations. I made sure to not select augmentation techniques like cropping that carry risk of erasing some objects from the image. Erasing objects from images is problematic as it would confuse the model about the actual cardinality of objects in the image. The transformations that were applied are shown in Table 6.2.

Table 6.2: Image transformations applied for ViT input preprocessing and augmentation.

Transformation	**Parameters / Notes**
Color Jitter	brightness=0.2, contrast=0.2, saturation=0.2, hue=0.05
Random Affine	rotation $\pm 10°$, scale=(0.95, 1.05)
Random Horizontal Flip	probability = 0.3
Random Vertical Flip	probability = 0.2
Gaussian Blur	kernel size=3, sigma=(0.1, 1.0)

A 80/20 train-validation split was performed on the CLEVR dataset, and the dataset augmentation was done on the training split to yield the final training set with 1200 samples. The final training set comprises 400 original samples and 800 augmented samples, totaling 1,200 examples. The numerosity distribution of the training dataset was kept balanced.

The hyperparameters for ViT finetuning were selected using heuristics and trial-and-error experimentation. Table 6.3 shows the hyperparameters the selected hyperparameters for finetuning of ViT.

Hyperparameter	**Value**
Number of epochs	12
Learning rate	1×10^{-4}
Weight decay	0.0001
Mixed precision training	True
Warmup steps	400
Early stopping patience	3
Target modules	MLP, att_v, att_q
Target layers	20–23

Table 6.3: Hyperparameters used for shallow LoRA fine-tuning of the ViT model on the selected target modules.

The learning rate set to a conservative value for a stable training process and a small amount of weight decay was used to help prevent catastrophic forgetting. The rationale behind the targe module and target layer selections can be found in the Methods section.

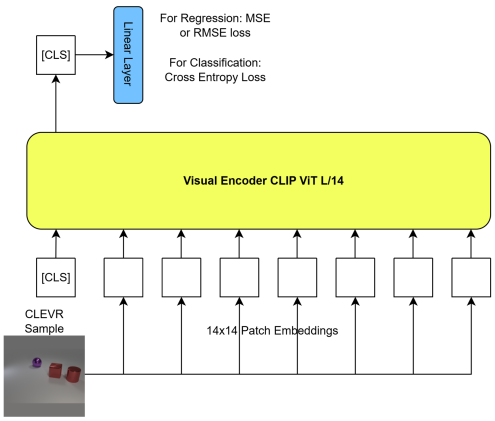


Figure 6.2: The Visual Encoder Tuning Setting: The gradients enter the attention layers through the CLS token

6.5 Finetuning the Entire LLaVA Pipeline (After Finetuning ViT)

After finetuning ViT, I finetuned the multimodal projector with the same hyperparameters as in section 6.3 to facilitate a fair comparison of how better image embeddings effect the outcomes of training.

6.6 The Numerical Understanding of LLaVA: Testing Further

Until this section, the focus has been on prompts that require the model to generate precise numerical responses. To test the numerical reasoning abilities better, I also run considered prompting LLaVA with a binary task. For the binary task, the prompt asked LLaVA to compare the number of objects with a pre-defined number. The prompts that were used for this experiments are as following:

```
USER: <image>\nAre there more than D objects in this image? Please
answer with yes or no.\nASSISTANT:
```

```
USER: <image>\nAre there less than D objects in this image? Please
answer with yes or no.\nASSISTANT:
```

The number D is selected for each (image,prompt) pair as either GT + diff or GT - diff where GT is the ground truth cardinality of the objects in the image. The model is constrained to answer with either 'yes' or 'no'. For each question, the probability of selecting either GT + diff or GT - diff is 0.5.

I try with diff=2 and diff=3 and generate 2 prompt queries each image. The experiment is run on 3 models: the baseline LLaVA, LLaVA with multimodal projector tuned and LLaVA with both visual encoder and the multimodal projector tuned.

6.7 Ablations

The ablations are run are as the following:

1. **Ablation study 1:** Running the multimodal projector training with LoRA R changed to 8 and the learning rate increased to 3×10^{-5}. In this ablation, I explored the possibility that a lower LoRA rank combined with a higher learning rate could increase generalization across different datasets and object categories.

2.**Ablation study 2:** In this ablation I explored whether in addition to the multimodal projector, unfreezing first few and last few attention layers of the language model could improve the performance. I similarly injected QLoRA adapters into these layers taking other hyperparameters from Table 6.1.

6.8 Performance Metrics

To evaluate the visual enumeration performance of LLaVA, I adopt several metrics that quantify both exact and approximate correctness of predictions, as well how much the predictions deviate from the ground truth.

6.8.1 Accuracy

Accuracy measures the proportion of correct predictions over the total number of samples. Given predicted outputs $\hat{y}_i$ and ground-truth labels y_i for N samples, I compute accuracy as:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} 1(\hat{y}_i = y_i), \tag{6.1}$$

where $1(\cdot)$ is the indicator function returning 1 if the condition is true and 0 otherwise. Accuracy is of central importance in this work.

6.8.2 Tolerant Accuracy

Tolerant accuracy relaxes the exact-match requirement by allowing predictions that are within a certain tolerance τ of the ground truth. For this work, the τ is set to 1 for the entire discussion. It is computed as:

$$\text{Tolerant Accuracy} = \frac{1}{N}\sum_{i=1}^{N} 1(|\hat{y}_i - y_i| \leq \tau). \tag{6.2}$$

6.8.3 Normalized Absolute Error (NAE)

The normalized absolute error (NAE) quantifies the average relative deviation between predicted and true values. For N samples, NAE is computed as:

$$\text{NAE} = \frac{1}{N}\sum_{i=1}^{N} \frac{|\hat{y}_i - y_i|}{y_i}, \tag{6.3}$$

where $\hat{y}_i$ is the predicted value and y_i is the corresponding ground-truth value. NAE is particularly useful in measuring error for counting experiments as it better accounts for the effects of Weber's Law [9].

These metrics together provide a comprehensive view LLaVA's performance, balancing exact correctness, approximate correctness, and relative deviation from ground truth.

7 Results

7.1 Selecting the Prompting Strategy

Before starting with the experiments that aim to improve visual enumeration of LLaVA, a prompt selection procedure was completed. All of the prompts that I consider constrain LLaVA to generate a numerical answer, excluding expressions like 'a few' and 'several'. The rationale behind this experiment is to see if different phrasing in the prompts create a substantial difference in the visual enumeration performance. If one of the prompts yields a notably better result, it will be selected for use in all of the consecutive experiments.

The prompts that are tested are shown below.

```
Prompt 1:
Count the total number of distinct objects in the photo.
Answer only with the final count as a numeral.

Prompt 2:
How many objects are there in the image? Respond only with the number.

Prompt 3:
Please count all apples visible in the image and reply with just the number.
```

```
Prompt 4:
Determine how many separate objects are in the photo. Only provide
the numeral.

Prompt 5:
Give the total count of distinct items in this picture. Only reply with
the number.
```

I have tested all of the prompts on the CCNL1PromptTest dataset. This dataset partition was used exclusively for the prompt selection procedure. A more detailed description of this partition can be found in the Methods section. Overall accuracy of the visual enumeration for each prompt is shown in Table 7.1.

Prompt	Test Accuracy
Count the total number of distinct objects in the photo. Answer only with the final count as a numeral.	0.3699
How many objects are there in the image? Respond only with the number.	0.4098
Please count all apples visible in the image and reply with just the number.	0.3979
Determine how many separate objects are in the photo. Only provide the numeral.	0.3460
Give the total count of distinct items in this picture. Only reply with the number.	0.3526

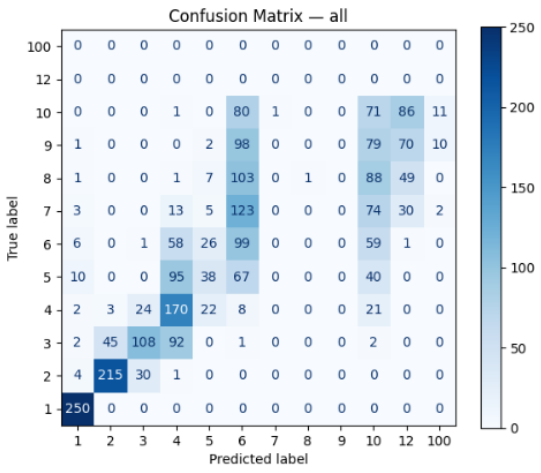
Table 7.1: Test accuracy for different prompt formulations.

While the differences between the test set accuracy do not vary drastically, the 2nd prompt achieved the highest accuracy and will be used for all of the subsequent experiments.

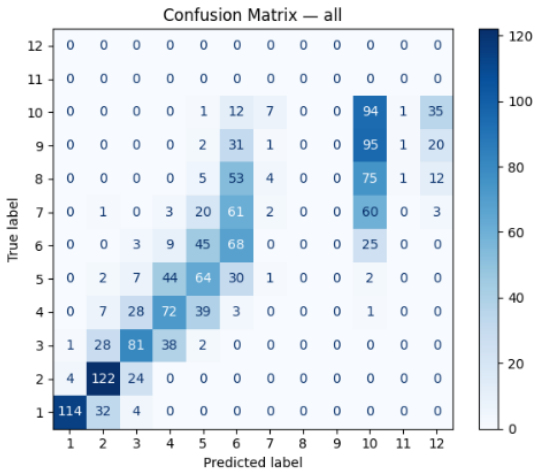
7.2 Baseline Results Without Finetuning

The visual enumeration accuracy for baseline LLaVA on the datasets CCNL1 and CCNL2 (that also includes CCNL2Test) are shown in Table 7.2. The confusion matrices for the datasets CCNL1 and CCNL2Test are shown in Figure 7.1. The confusion matrices reveal that LLaVA has accuracy around 40 percent for the visual enumeration task. As expected, the performance is in general better for lower numbers than higher numbers. In addition, LLaVA seems to perform better on certain categories. An example of such an object category is the 'apples' category

of the CCNL2 dataset—the only object category on which baseline LLaVA achieves an accuracy higher than 50 percent. The likely reasons for better performance on 'apples' are more uniform object shapes and better contrast between the object and the background.



(a) Confusion Matrix on CCNL1 dataset untuned LLaVA



(b) Confusion Matrix on CCNL2 test dataset untuned LLaVA

Figure 7.1: Confusion Matrices for untuned baseline LLaVA

Category	Accuracy	Tol. Acc.	NAE
apples	0.3900	0.6400	0.1734
butterflies	0.3720	0.6960	0.1762
dots	0.3510	0.5216	0.7752
fastcards	0.4080	0.6220	0.2028
people	0.3760	0.6280	0.1978
CCNL1 Avg.	**0.3793**	**0.6211**	**0.3069**

(a) Performance Metrics (CCNL1)

Category	Accuracy	Tol. Acc.	NAE
bottles	0.4260	0.7300	0.1916
buttons	0.4240	0.7560	0.1608
knives	0.3840	0.6940	0.2067
CCNL2Test Avg.	**0.4113**	**0.7267**	**0.1864**
apples	0.5340	0.8340	0.1055
cups	0.3280	0.6480	0.2563
CCNL2 All Avg.	**0.4192**	**0.7324**	**0.1842**

(b) Performance Metrics (CCNL2Test)

Table 7.2: Comparison of Performance Metrics Across Datasets

Another interesting observation is that LLaVA struggles with enumerating images that contain 7,8 or 9 objects. Instead of giving the correct answer, LLaVA lumps all the predictions into numbers 6 and 10. This points out to an absence of meaningful internal representation for numerosities of 7, 8 and 9 causing LLaVA to default to the 'boundary' numerosities 6 and 10.

To understand this phenomenon better, I have plotted the occurrence frequency of different numerosities (both word format and numeric format) in the CC3M595K image-caption

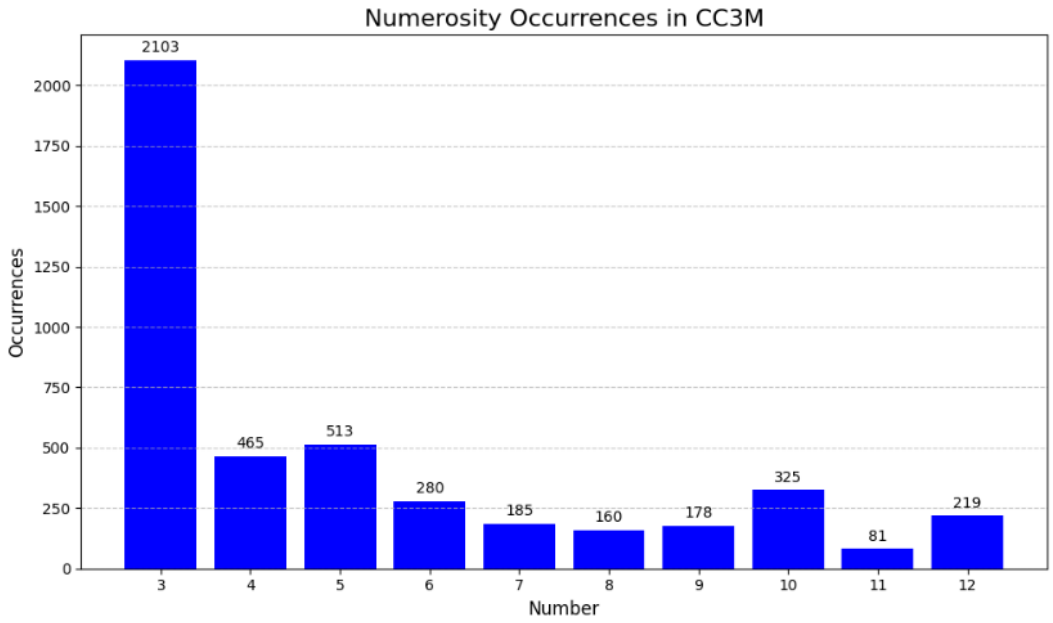


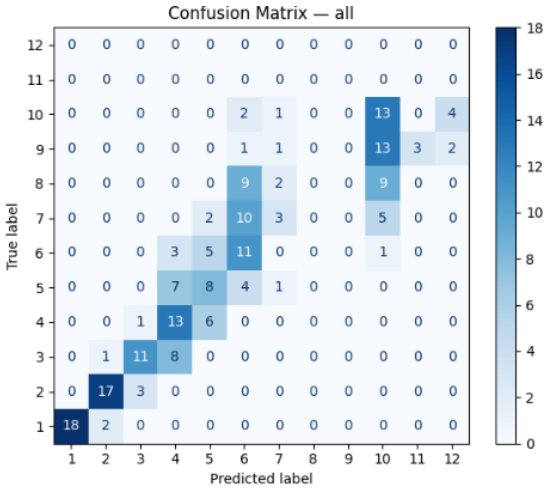
Figure 7.2: Occurrence of Numerosities in the Captions of CC3M595K - Less occurance of 7, 8 and 9 with respect to other numerosities

dataset that was used for image-text alignment pretraining in Figure 7.2. The plot shows that there exists a 'dip of appearance' for the counts of 7, 8, and 9. This dip possibly explains why LLaVA struggles so much with predicting numerosities 7, 8, and 9.

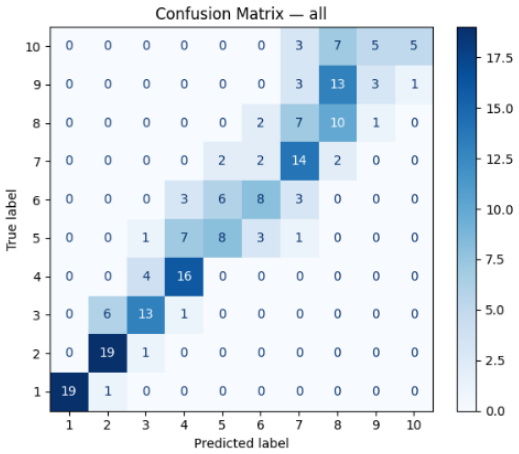
7.3 Finetuning and Testing Without Partitioning Numerosities or Categories

When evaluated on a test set matching the training distribution, the finetuned LLaVA model showed substantial accuracy gains. Figure 7.3 shows the overall confusion matrices and table 7.3 summarizes the test set accuracies across object types:

The tables and the confusion matrices show that in absence of a distribution shift, the improvement in visual enumeration accuracy is significant. Will these substantial improvements transfer when we change the test set categories?



(a) Baseline LLaVA confusion matrix (CCNL2Train 20% split)



(b) Finetuned LLaVA confusion matrix (CCNL2Train 20% split)

Figure 7.3: Comparison of confusion matrices between untuned (left) and finetuned (right) LLaVA models.

Category	Accuracy	Tol. Acc.	NAE
apples	0.4706	0.7549	0.1540
cups	0.4694	0.8061	0.1201
average	0.4700	0.7800	0.1374

(a) Untuned LLaVA performance on 20 percent random split of CCNL2Train

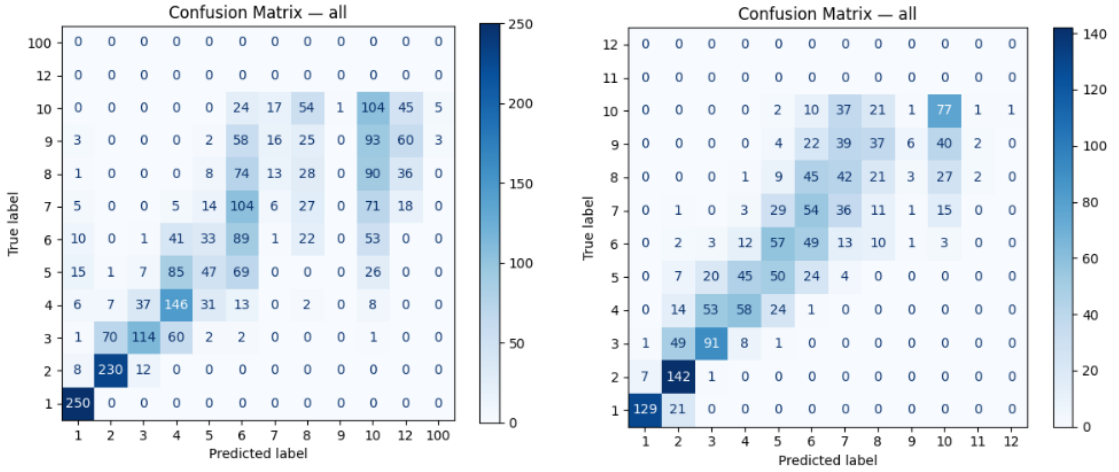
Category	Accuracy	Tol. Acc.	NAE
apples	0.5882	0.9118	0.0883
cups	0.5612	0.8673	0.0897
average	0.5750	0.8900	0.0890

(b) Finetuned LLaVA performance on 20 percent random split of CCNL2Train

Table 7.3: Side-by-side comparison of untuned (left) and finetuned (right) LLaVA numerosity performance on selected CCNL2 categories.

7.4 Finetuning the Multimodal Projector (Before Finetuning ViT)

The next step is to apply QLoRA to the multimodal projector of LLaVA. The entire pipeline was trained on the CCNL2Train dataset. The performance of LLaVA for the exact counting task after finetuning the multimodal projector are shown in Table 7.4. The confusion matrices are shown in Figure 7.4.



(a) Confusion Matrix on CCNL1 dataset (Only the projector tuned)

(b) Confusion Matrix on CCNL2 test dataset (Only the projector tuned)

Figure 7.4: Confusion Matrices for LLaVA with tuned multimodal projector - the improvements for CCNL2Test are more prominent than CCNL1

Category	Accuracy	Tol. Acc.	NAE
apples	0.3840	0.6680	0.1586
butterflies	0.4180	0.7640	0.1399
dots	0.3784	0.5667	0.4382
fastcards	0.4360	0.6500	0.1825
people	0.4040	0.7060	0.1812
average	0.4040	0.6705	0.2210

(a) Performance Metrics (CCLN1)

Category	Accuracy	Tol. Acc.	NAE
bottles	0.5080	0.8260	0.1187
buttons	0.3840	0.7060	0.1673
knives	0.4260	0.7680	0.1661
average	0.4393	0.7667	0.1507

(b) Performance Metrics (CCLN2Test)

Table 7.4: Side-by-side comparison of performance metrics for CCLN1 (a) and CCLN2Test (b).

The category-wise accuracy table and the confusion matrices reveal a modest 3 percent improvement for both CCNL2Test and CCNL1 datasets. Another notable improvement is that

the model started to started predicting 7, 8 and 9. The accuracy for these numerosities is still quite low, but there is still improvement compared to the baseline model.

In addition to measuring improvement in the visual enumeration task across different datasets and across different object types, it is interesting to test how good the visual enumeration transfer when we train on only on even numbers and test only on the odd numbers. To test this, the multimodal projector was trained on CCNL2TrainEven dataset and tested on the CCNL2TrainOdd dataset. The test set metrics are shown in Table 7.5 and the confusion matrix for odd number is shown in Figure 7.5.

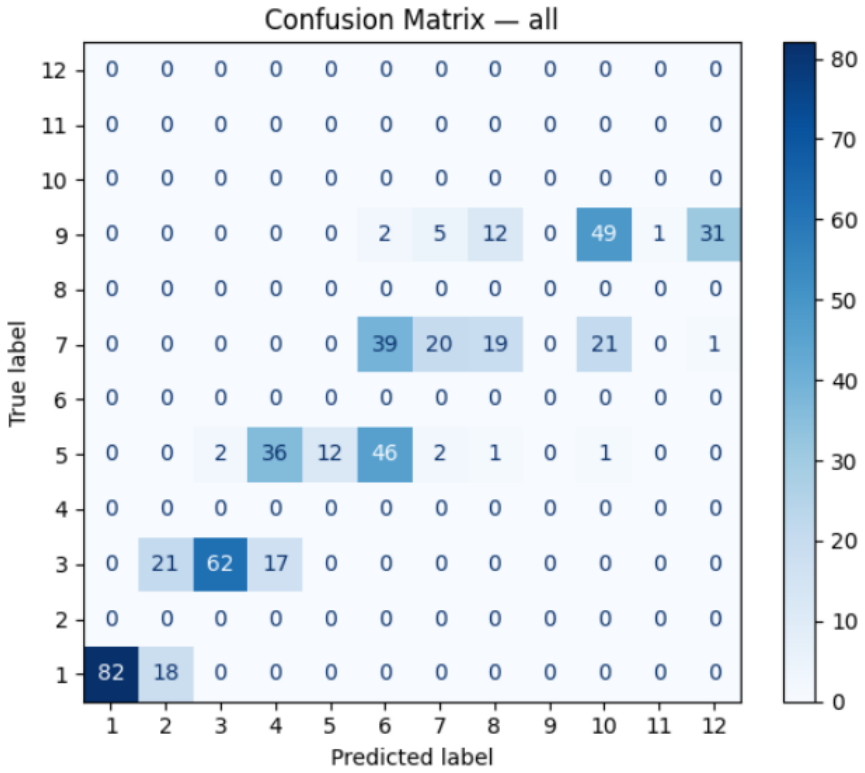


Figure 7.5: The confusion matrix for odd numbers CCNL2Train

Category	Accuracy	Tol. Acc.	NAE
baseline	0.3600	0.8000	0.2063
projector tuned	0.3520	0.8660	0.1748

Table 7.5: Performance Improvement on Odd Numbers After Tuning Only on Even Numbers

The results suggest almost no transfer from even to odd numbers. This suggests that when only the multimodal projector is tuned, the model does not get better at predicting the nu-

merosities that did not appear during the training phase.

7.5 ViT Finetuning

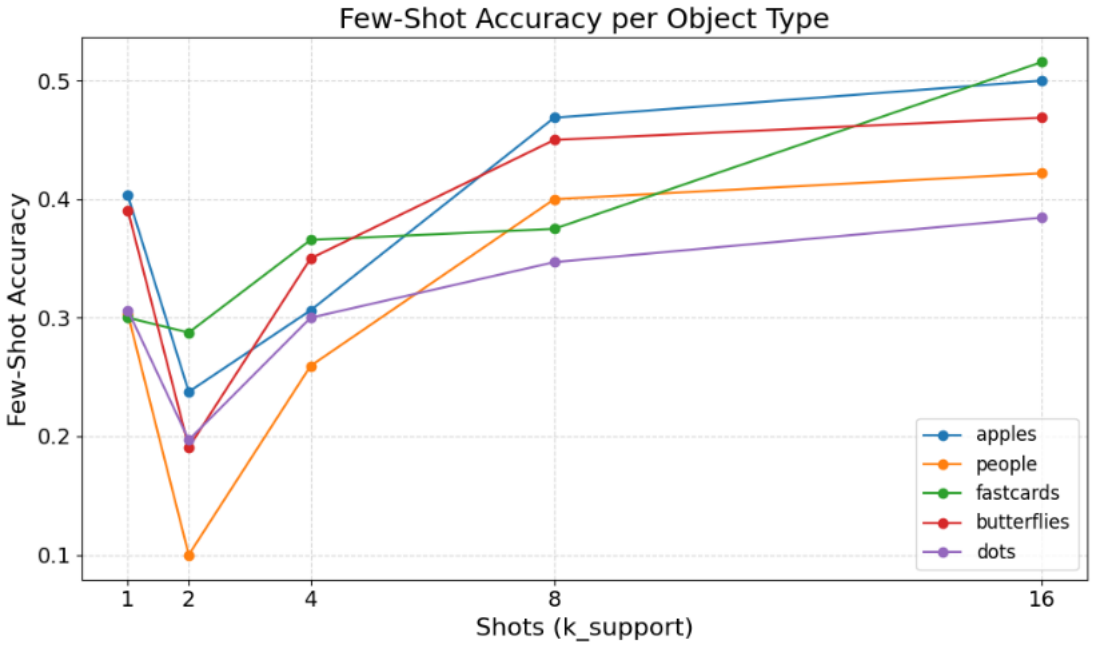
After I demonstrated that tuning only the multimodal projector does not provide substantial transfer for counting task across distribution shift, the next step is to check whether we can do better with better image representations. Starting with few-shot linear probing experiments on the representations, Figure 7.6 shows the few-shot accuracies for each object category in the CCNL1 and CCNL2Test datasets.

The results show that LLaVA performs better in few-shots setting for the CCNL1 dataset. This makes sense because CCNL1 dataset contains 2-dimensional images that contain uniformly white background. Learning to classify images with less background and lighting variation is easier for a linear classifier. As a result, it is expected that a linear-probe would achieve a better accuracy on CCNL1. Another interesting observation is that for CCNL2 dataset, the linear classifier comparatively performed better on "apples" compared to other categories. After a brief inspection of the CCNL2 dataset I noted that images containing 'apples' category seem to have more uniform background and more contrast between the objects (apples) and the background. This observation is in parallel with observation that baseline LLaVA performed the best on 'apples' among other categories of CCNL2.

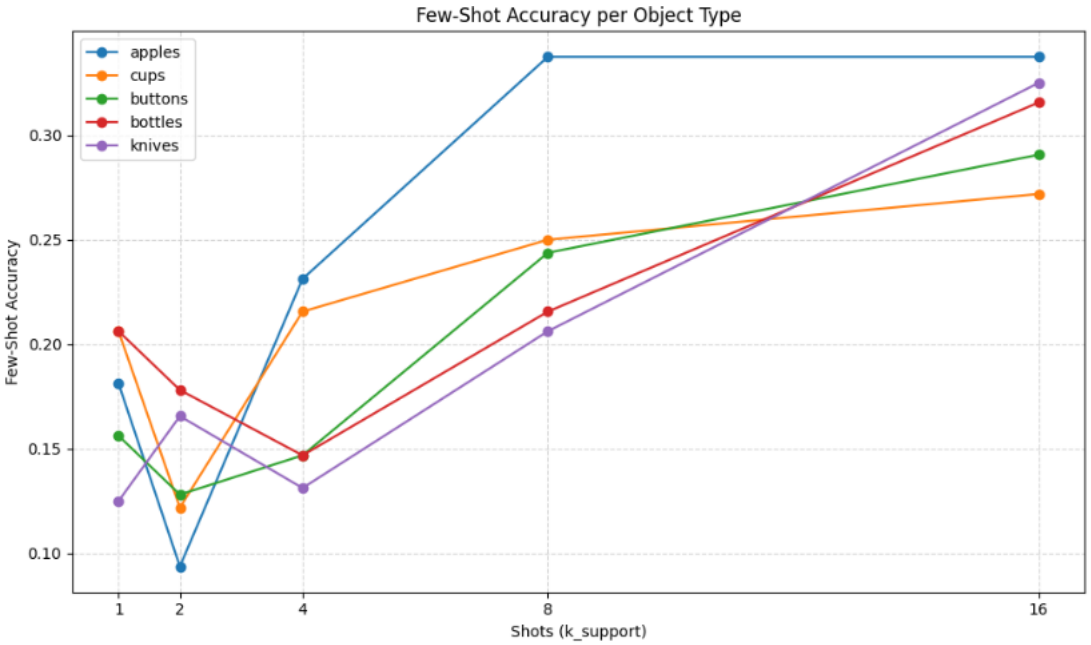
After tuning ViT, few shot performance improved consistently across all of the object types. Figure 7.7 demonstrates category-wise accuracy values for few-shot setting while figure 7.8 shows the comparison of overall few-shot accuracy between tuned and baseline ViT.

Figure 7.8 reveals a noticeable improvement in few-shot performance after finetuning. Besides the constant upward trend after 2-shots up to 16-shots, a significant accuracy dip can be observed at 2-shots. At first, a decrease in accuracy for 2-shot with respect to 1-shot seems counterintuitive. A possible explanation is that providing 2 examples per category confuses the linear classifier due to these 2 samples possibly having different object placements, lighting conditions and background. After increasing the number of shots further to 4 and beyond, the model starts to see enough diverse samples to start extracting meaningful patterns to predict the numerosity better.

To understand better how the image embeddings have changed after the finetuning procedure, I have reduced the embeddings to 2-dimensions with Principal Component Analysis (PCA). Figure 7.9 displays the embeddings for baseline ViT and tuned ViT. In both figures, I analyzed the embeddings of all categories of CCNL2 dataset.

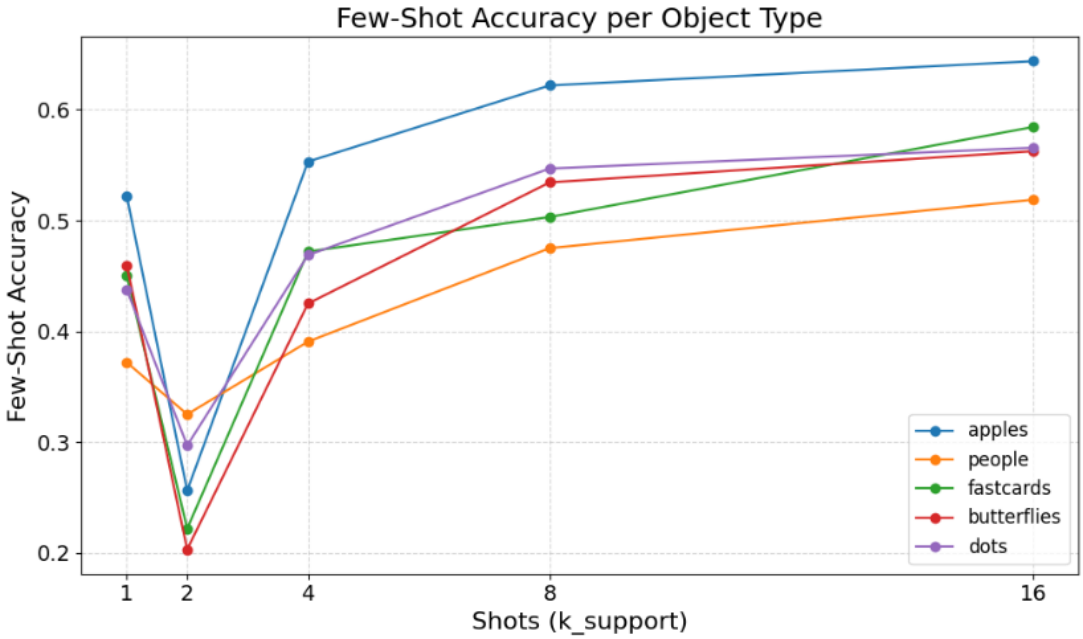


(a) CCNL1 Base ViT Few-Shot Results

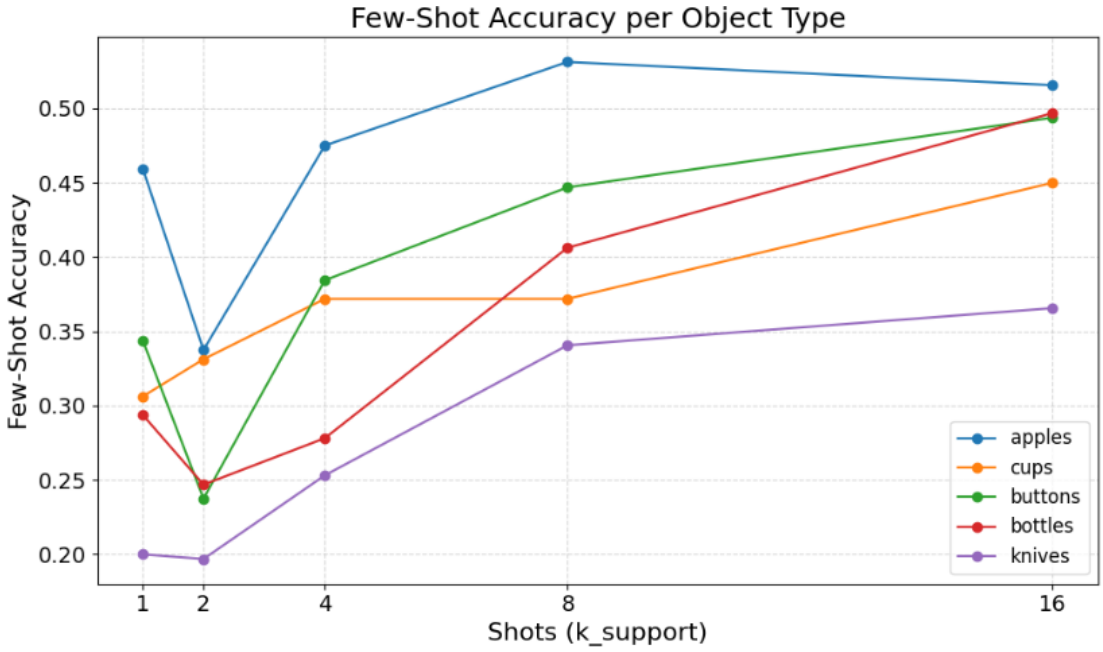


(b) CCNL2 Test ViT Few-Shot Results

Figure 7.6: Few-shot performance of the baseline ViT model on the CCNL1 and CCNL2 datasets.



(a) CCNL1 Tuned ViT Few-Shot Results



(b) CCNL2 Test Tuned ViT Few-Shot Results

Figure 7.7: Few-shot performance of the tuned ViT model on the CCNL1 and CCNL2 datasets.

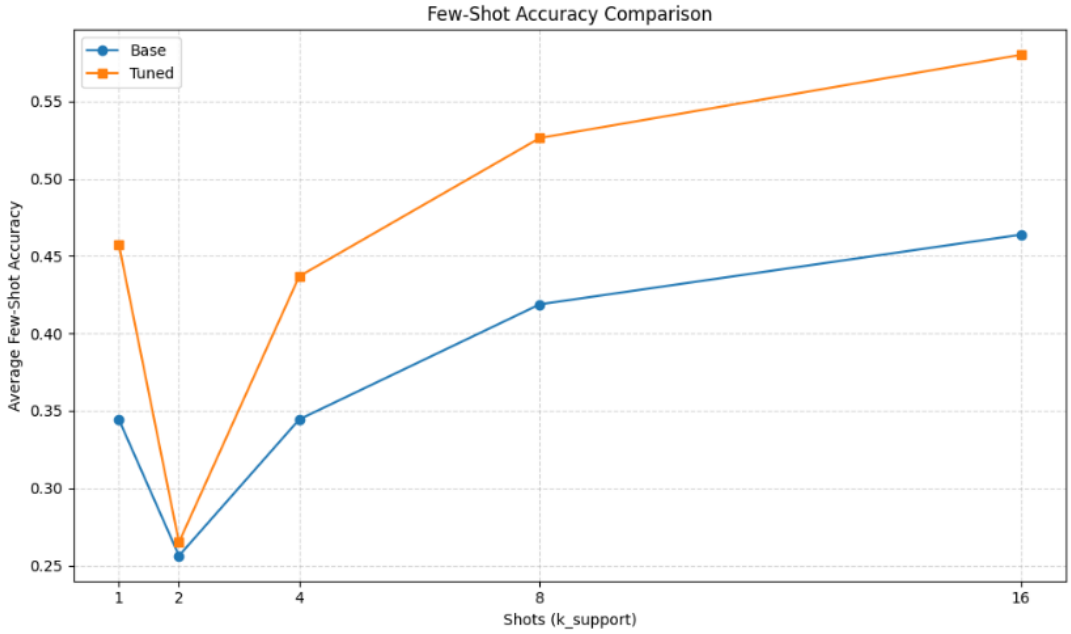
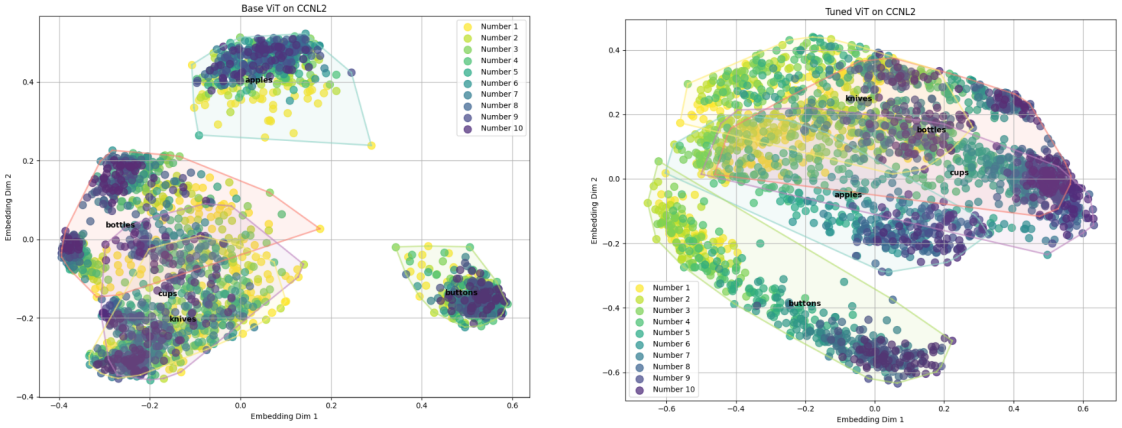


Figure 7.8: Comparison of few-shot performance between the baseline ViT and tuned ViT on the CCNL2 dataset.



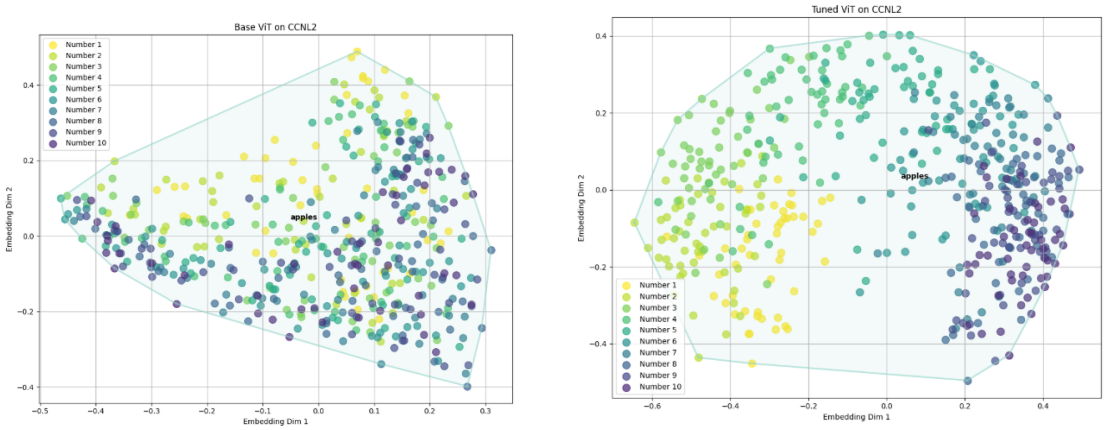
(a) Embeddings for CCNL2 (Baseline ViT)

(b) Embeddings for CCNL2 (Tuned ViT)

Figure 7.9: Comparison of image embeddings obtained from the baseline ViT and the tuned ViT on the CCNL2 dataset.

In the PCA projections, a notable pattern emerges: while the baseline ViT embeddings show no obvious structure, the embeddings of the tuned ViT align along a clear direction corresponding to numerosity, suggesting that tuning captures information relevant to object count.

The PCA plots clearly demonstrate that while the embeddings are aligned in one dominant PCA direction according to their numerosity, there exists an orthogonal dimension along which the semantic content of the image varies. This suggests that tuned ViT may encode the semantic content of the image and the number of objects in the image in orthogonal directions. A stronger claim for orthogonality or independence of numerosity and semantical content warrants a more detailed analysis but the PCA analysis suggests that ViT started to encode images according to count, creating more numerosity-aware embeddings. To examine this more closely, I applied PCA dimensionality reduction to a single category: apples. Figure 7.10 shows the dimensionality-reduced embeddings of both the baseline ViT and the tuned ViT for the apples category of the CCNL2 dataset.

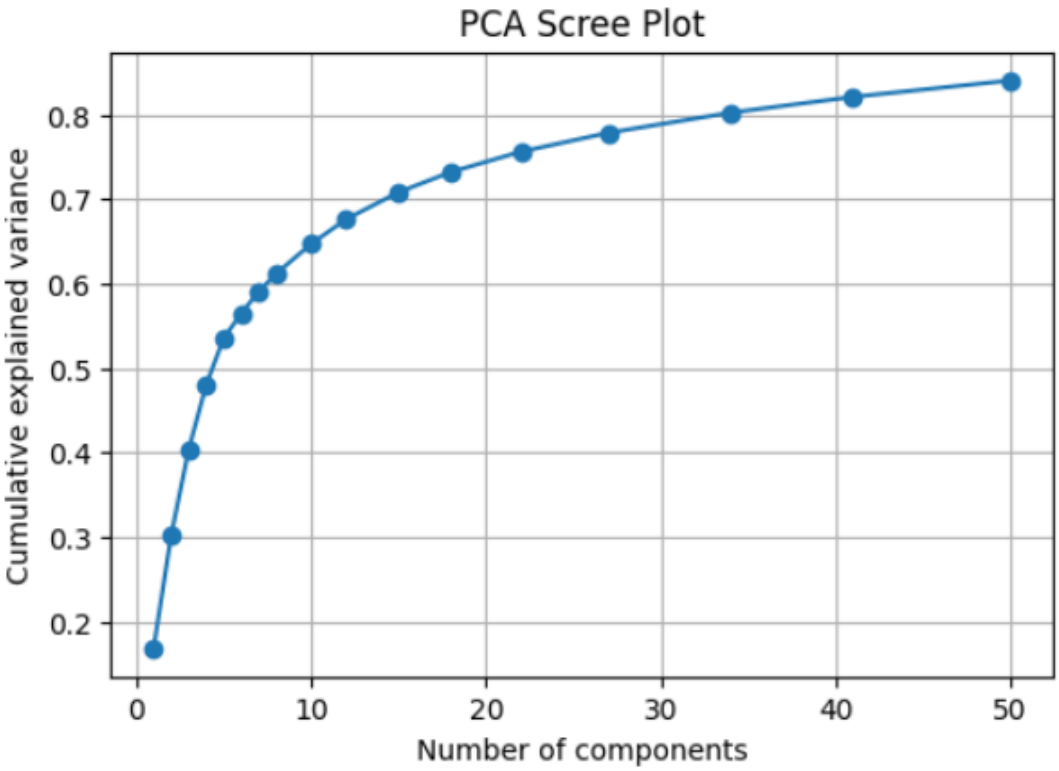


(a) Embeddings for CCNL2-Apples (Baseline ViT)

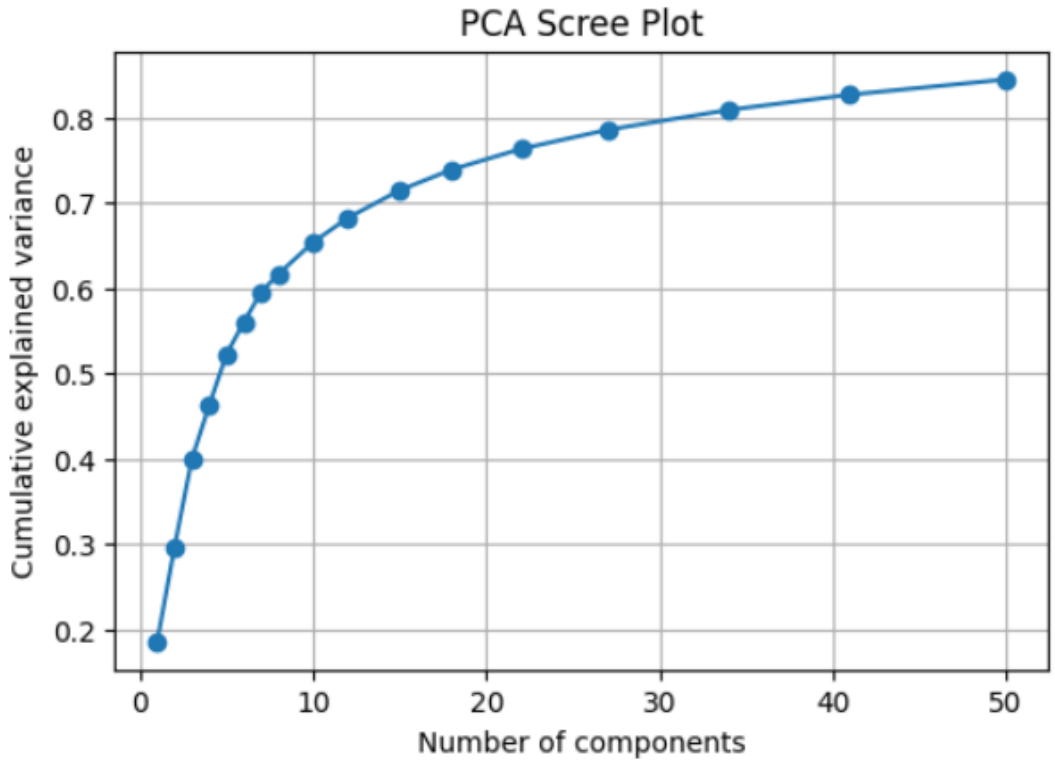
(b) Embeddings for CCNL2-Apples (Tuned ViT)

Figure 7.10: Comparison of image embeddings obtained from the baseline ViT and the tuned ViT on the CCNL2 dataset.

These plots reinforce the claim that tuning made ViT embeddings notably more numerosity-aware. The improved few-shot performance also supports this: the numerosity embeddings are encoded in a more linearly separated way, increasing the classification accuracy that can be achieved using linear decision boundaries. A more detailed analysis is called for to determine if the embeddings are organized truly according to the abstract 'numerosity' quantity or other continuous variables like object area, contour length and object density also play a significant role. It is also useful to look at the scree plots for PCA dimensionality reduction for each of the embeddings. The scree plots are displayed in Figure 7.11.



(a) The Scree Plot for the Baseline ViT CLS Embeddings



(b) The Scree Plot for the Tuned ViT CLS Embeddings

Figure 7.11: Scree plots for embeddings of the baseline ViT and the embeddings of the tuned ViT.

The plots show that about 30 percent of variance is captured if I keep 2-dimensions. Even though this makes it harder to rely only on plots to reason about linear separability, the few-shot linear probing improvements support that the embeddings have become more linearly separable.

7.6 Finetuning the Multimodal Projector (After Finetuning ViT)

After enhancing the vision transformer with more linearly separable numerosity embeddings, I have tuned the multimodal projector again to check if more linearly separable embeddings make LLaVA more robust to distribution shifts.

Category	Baseline			MM Tuned			MM + ViT Tuned		
	Acc	Tol. Acc	NAE	Acc	Tol. Acc	NAE	Acc	Tol. Acc	NAE
CCNL1									
apples	0.3900	0.6400	0.1734	0.3840	0.6680	0.1586	0.4000	0.6880	0.1488
butterflies	0.3720	0.6960	0.1762	0.4180	0.7640	0.1399	0.4860	0.8780	0.1144
dots	0.3510	0.5216	0.7752	0.3784	0.5667	0.4382	0.3804	0.5608	0.6890
fastcards	0.4080	0.6220	0.2028	0.4360	0.6500	0.1825	0.4540	0.6700	0.1741
people	0.3760	0.6280	0.1978	0.4040	0.7060	0.1812	0.3460	0.5860	0.2314
CCNL1 Avg.	0.3793	0.6211	0.3069	0.4040	0.6705	0.2210	0.4131	0.6761	0.2732
CCNL2Test									
bottles	0.4260	0.7300	0.1916	0.5080	0.8260	0.1187	0.5680	0.8480	0.1081
buttons	0.4240	0.7560	0.1608	0.3840	0.7060	0.1673	0.4720	0.8340	0.1268
knives	0.3840	0.6940	0.2067	0.4260	0.7680	0.1661	0.4260	0.7840	0.1617
CCNL2 Test	0.4192	0.7324	0.1842	0.4393	0.7667	0.1507	0.4887	0.8220	0.1322

Table 7.6: Comparison of Baseline, MM Tuned, and MM + ViT Tuned performance across CCNL1 and CCNL2 datasets. Green values indicate the highest Accuracy per row.

Table 7.6 shows the accuracy for the three finetuning scenarios I have considered so far. While tuning only the multimodal projector provided a modest improvement (mostly a few percent), better linearly separated image embeddings resulted in a better transfer across object categories. It is now interesting to ask: how much finetuning ViT improves transfer between numerosities? Table 7.7 shows the improvement of accuracy for the same even-odd training scenario considered earlier.

Category	Accuracy	Tol. Acc.	NAE
baseline	0.3600	0.8000	0.2063
projector tuned	0.3520	0.8660	0.1748
projector + vit tuned	0.4020	0.9740	0.1397

Table 7.7: Performance Comparison Across Baseline, Projector-Tuned, and Projector + ViT-Tuned Models

Table 7.7 reveals that LLaVA with a tuned ViT has better visual enumeration performance for images with even numbers after being tuned only on odd numbers. This was not the case when I finetuned only the multimodal projector. My hypothesis is that the linear structure of the numerosity embeddings acquired during finetuning enables the model to interpolate from even to odd numbers. Before the finetuning procedure, the numerosity embeddings did not have an ordering pattern, and LLaVA could not exploit any structure to improve visual enumeration ability on numerosities that it had not seen during training. In short, a structural prior of the image embeddings resulted in much better generalization ability.

7.7 Ablations

7.7.1 Finetuning the Language Model

I investigated how finetuning the vision encoder affects LLaVA's visual enumeration performance. But can we improve counting performance even further by also tuning the language model? Table 7.8 displays the performance metrics on CCNL2Test for this setting. During training, in addition to the multimodal encoder, the first and last few layers of the language model were also unfrozen.

Category	Accuracy	Tol. Acc.	NAE	Tuning
bottles	0.4280	0.6340	0.2164	MM Projector + LM + ViT
buttons	0.4500	0.7860	0.1555	MM Projector + LM + ViT
knives	0.3520	0.6520	0.2153	MM Projector + LM + ViT
All	0.4100	0.6907	0.1958	MM Projector + LM + ViT
bottles	0.5680	0.8480	0.1081	MM Projector + ViT Tuned
buttons	0.4720	0.8340	0.1268	MM Projector + ViT Tuned
knives	0.4260	0.7840	0.1617	MM Projector + ViT Tuned
All	0.4887	0.8220	0.1322	MM Projector + ViT Tuned

Table 7.8: Performance Metrics for CCNL2Test dataset under different tuning configurations.

The results reveal that unfreezing the language model reversed all of the performance gains that were observed after tuning only the multimdal projector. I hypothesize that this is because the attention layers the language model have a high expressive power and tuning them on CCNL2Train dataset causes an overfit to the exact categories it has seen during the training, canceling any improvements for the object categories the model has not seen during the training.

7.7.2 Effect of a Higher Learning Rate on Projector Tuning with a Tuned ViT

Tuning the ViT led to improved counting accuracy within the same dataset and across different object categories, nevertheless the gains observed in cross-dataset evaluations were pretty small. To see if better cross-dataset improvements in accuracy are possible by increasing the learning rate, I increased the learning rate and retrained the entire LLaVA pipeline (again only the multimodal projector is unfrozen). The accuracy values of the trained model for CCNL2Test and CCNL1 datasets are shown in Table 7.9 .

Category	**Old LR**			**Increased LR**		
	Acc	Tol. Acc	NAE	Acc	Tol. Acc	NAE
CCNL1						
fastcards	0.4540	0.6700	0.1741	0.5160	0.7220	0.1431
people	0.3460	0.5860	0.2314	0.3880	0.6540	0.1988
dots	0.3804	0.5608	0.6890	0.4294	0.6176	0.4269
butterflies	0.4860	0.8780	0.1144	0.5060	0.8400	0.1165
apples	0.4000	0.6880	0.1488	0.4000	0.6540	0.1586
All	0.4131	0.6761	0.2732	0.4478	0.6972	0.2097
CCNL2 Test						
bottles	0.5680	0.8480	0.1081	0.5800	0.8920	0.0948
buttons	0.4720	0.8340	0.1268	0.4640	0.8020	0.1310
knives	0.4260	0.7840	0.1617	0.4220	0.7740	0.1618
All	0.4887	0.8220	0.1322	0.4885	0.8227	0.1292

Table 7.9: Comparison of MM projector + ViT tuned performance on CCNL1 and CCNL2 Test datasets between the original learning rate and an increased learning rate.

The table reveals that a higher learning rate produced a better accuracy on CCNL1, while the accuracy values for CCNL2 dataset remained almost the same. This highlights that the

results in the previous experiments can be improved even further by a more extensive search of hyperparameters.

7.8 Transfer to A Different Numerical Task: Binary Decision Task

Lastly, I have conducted an experiment to see how the improvement in visual enumeration ability transferred to a relevant task that also requires numerical reasoning: a binary numerical decision task. For this, LLaVA was required to decide if there are more than a specified number of objects in the image. There are 2 possible answers: 'yes' or 'no'. I considered the prompts with diff=2 and diff=3 values on CCNL2Test dataset (totaling 3000 test samples) and the results are reported in the table 7.10.

Model / Condition	**Accuracy (diff = 2)**	**Accuracy (diff = 3)**
Rand. Chance	50.0	50.0
Base	57.0	60.43
MM + ViT tuned	59.4	65.8

Table 7.10: Accuracy for binary numerosity decision task at diff = 2 and diff = 3. A clear improvement for when ViT is finetuned.

The results clearly demonstrate that even though LLaVA was tuned on the exact visual enumeration task, the improvements transferred to other tasks requiring numerical understanding. This reinforces the claim that LLaVA not only learned to predict the exact counts, it also gained a better grasp of numerical quantities.

7.8.1 Summary of Findings

The results obtained demonstrate that a stronger vision encoder with better numerosity representations substantially improves performance on downstream visual enumeration tasks, indicating that weak numerosity discrimination in the visual backbone can act as a bottleneck for the enumeration abilities of VLMs and MMLLMs. Additionally, the findings show that synthetic data is highly effective in pretraining settings: pretraining on synthetic numerosity datasets markedly enhances downstream task performance, underscoring the value of controlled synthetic supervision for improving model robustness and numerical sensitivity.