

# Session 4 - Image Captioning

**Team 4:**

- **María José Millán**
- **Agustina Ghelfi**
- **Laila Aborizka**



# DATA EXPLORATION AND CLEANING

DATASET OVERVIEW: THE DATASET WAS FILTERED AS WE ONLY NEEDED (IMAGE NAME ,TITLE), OBTAINING 13501 IMAGES.



The word cloud shows the frequency of words in the titles, with "Salad" and "Chicken" appearing most frequently.

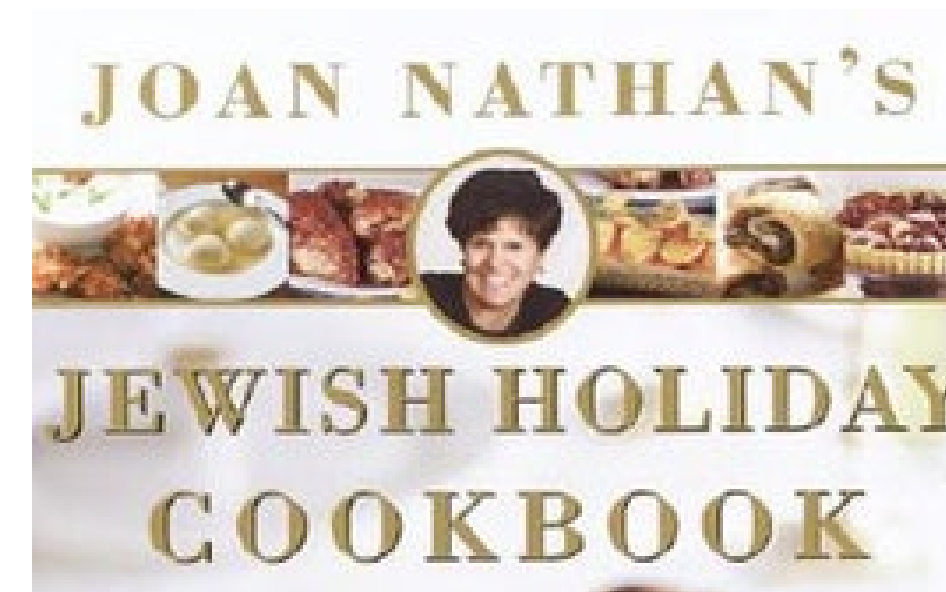
- Image Name : The filename of the image.
- Title : The caption or title describing the image.
- Data Cleaning Steps: 5 missing titles. Titles with missing values were removed and their respective images also.
- Non-Existent Images: Some entries had Image Name values that didn't match any actual image files. These rows were removed to ensure the dataset only contains valid image-caption pairs.

## NEW DISCOVERIES OF THIS WEEK

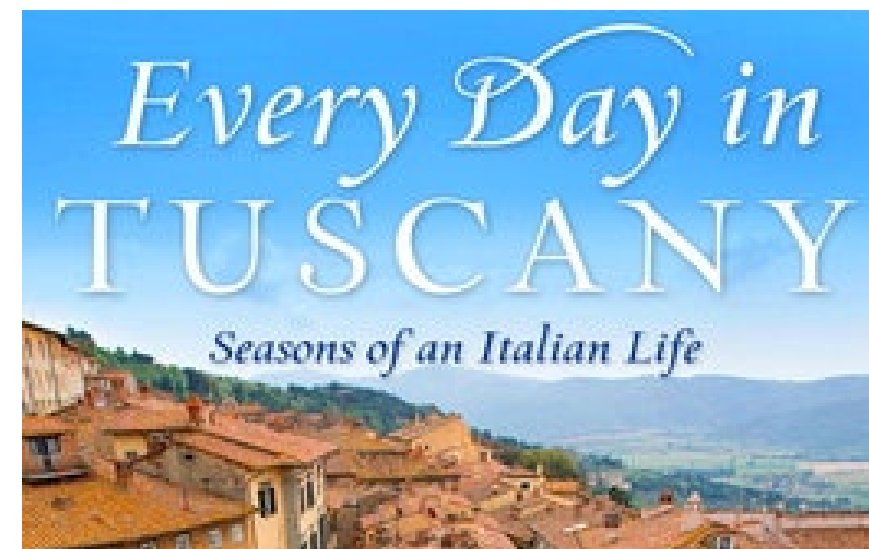
SOME IMAGES ARE NOT SPECIFIC OF THE FOOD DOMAIN AND  
OTHERS NOT HAVE A CLEAR CAPTION RELATED TO FOOD



# Zombies Rising



## Zamosc Gefilte Fish



# Zuppa di Cavolo Nero, Cannellini, e Salsicce: Kale, White Bean, and Sausage Soup



## Whipped Sweet Potatoes with Honey

# ARCHITECTURE OVERVIEW: ViT + GPT-2

To tackle the image captioning task, we used a baseline model consisting of a Vision Transformer (ViT) encoder and a GPT-2 decoder with word-piece level text representation [1].

## ENCODER - ViT

The encoder is a Vision Transformer (ViT) pretrained on ImageNet, which processes the input image and extracts rich visual features. The output consists of a set of feature vectors that summarize the image content.

### Embedding

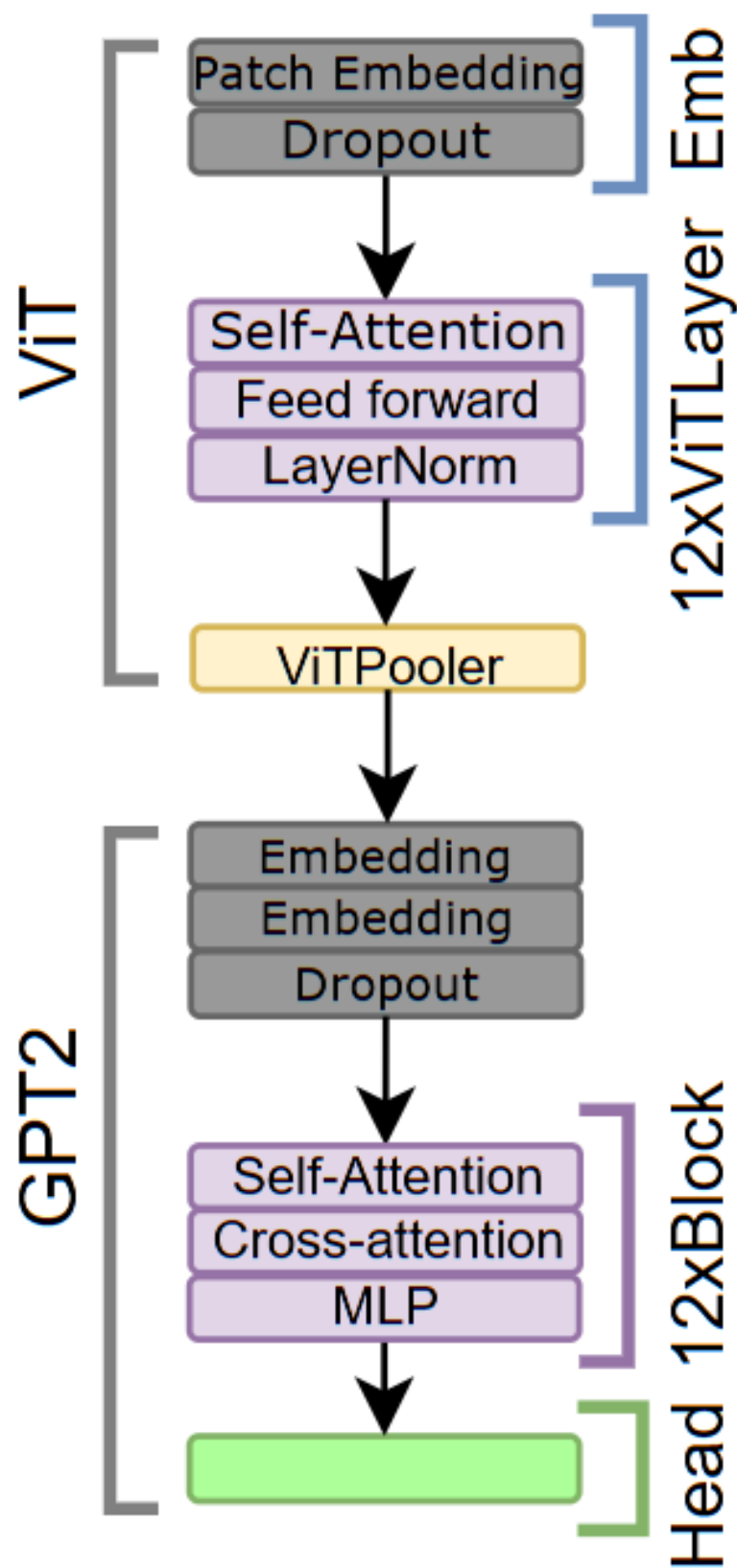
ViT processes images by dividing them into fixed-size patches. These patches are embedded into feature vectors using a 2D convolution, and the model uses self-attention to capture relationships between patches.

### 12 x ViT Layer

The core of ViT is its encoder, which includes 12 layers of self-attention and feedforward networks, allowing it to model spatial dependencies. Layer normalization is applied to stabilize training, and a pooling layer creates a global representation for tasks like classification.

### Feature projection

To adapt the visual features for caption generation, we applied a linear projection layer that maps the encoder's output into the appropriate input space for GPT-2, ensuring compatibility.



## DECODER - GPT-2

The decoder is a GPT-2 model pretrained on diverse text datasets, responsible for generating captions in a sequential manner. It generates text autoregressively, predicting the next token in the sequence based on prior context. The final output layer applies a softmax function to produce probability distributions over possible tokens.

### Embeddings

GPT-2 processes text by converting input tokens into dense embeddings, incorporating positional encodings to preserve word order. These embeddings serve as the foundation for sequential text generation.

### 12 x Block

The core of GPT-2 consists of 12 transformer blocks, each containing self-attention mechanisms followed by feedforward networks, with layer normalization applied for stability. Self-attention allows the model to capture dependencies across the entire sequence, ensuring contextual coherence in generated text. Dropout is applied within the blocks to improve generalization.



# TRAINING STRATEGY & EXPERIMENTS

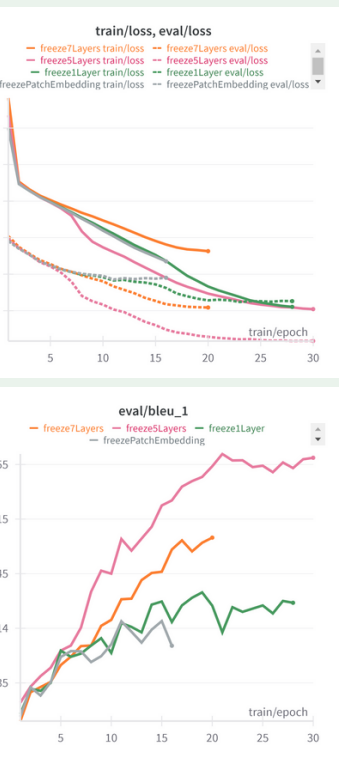
We first performed direct evaluation of the model using pretrained weights from huggingface. We then explored three training strategies: (1) fine-tuning ViT while keeping GPT-2 frozen, (2) freezing ViT and fine-tuning GPT-2, and (3) fine-tuning both ViT and GPT-2. These setups allowed us to analyze the impact of visual feature adaptation and language modeling on caption generation. Several experiments were conducted for each strategie, using different optimizers, learning rates and unfreezing layers one by one. The most relevant results are presented below.

## VIT FINE-TUNE + GPT-2 FREEZE



The figure on the left illustrates how training and validation loss vary when using different learning rate values, while keeping all other parameters fixed (AdamW optimizer, batch size = 32). Early stopping was applied with a patience of 5 and delta of 0.001.

This experiment was performed with all layers of ViT unfrozen. The model with learning rate 1e-5 was selected as the best in this case as the model with lr 1e-6 has higher loss values and presents some overfitting, and the model with lr 1e-4 though lower in loss, shows significant overfitting.



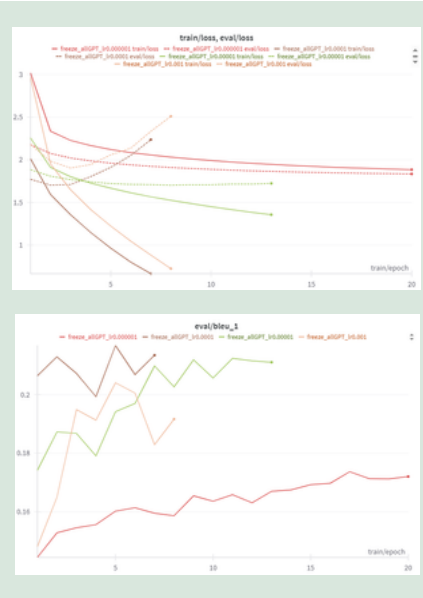
The figures on the left show the most relevant results from the experiments of freezing layers, one by one, of ViT, while keeping GPT-2 all frozen.

Looking at the loss figure, the models that stand out are the one with 5 blocks frozen (pink curves), as it has the lowest validation loss, and the one with only 1 block frozen (green curves), which has less overfitting than all the others.

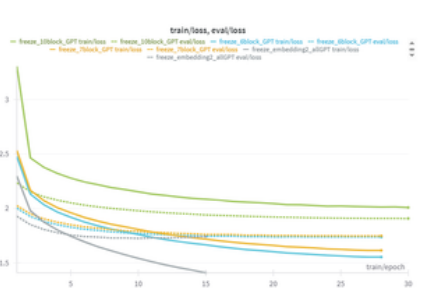
To make a decision about the best model we also analyzed the metrics BLEU-1, BLEU-2, ROUGE-L and METEOR over the validation set. The figure on the left shows only BLEU-1 as an example but the behavior among models was similar for all these metrics.

Taking into consideration all the metrics, including the loss, we can conclude that the best model is the one where the embedding layers and the ViT blocks were frozen up to block number 5.

## VIT FREEZE + GPT-2 FINE-TUNE



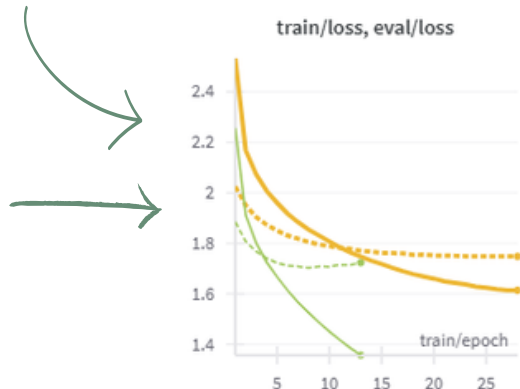
The models trained with learning rates of 1e-3 and 1e-4 show significant overfitting. The model with lr 1e-5 was selected as the best in this case over the model with lr 1e-6, as despite exhibiting slightly more overfitting, it achieves better performance in the BLEU-1, BLEU-2, ROUGE-L and METEOR metrics (only BLEU-1 showed here as an example). This suggests a better balance between learning capacity and generalization, making it the preferred option.



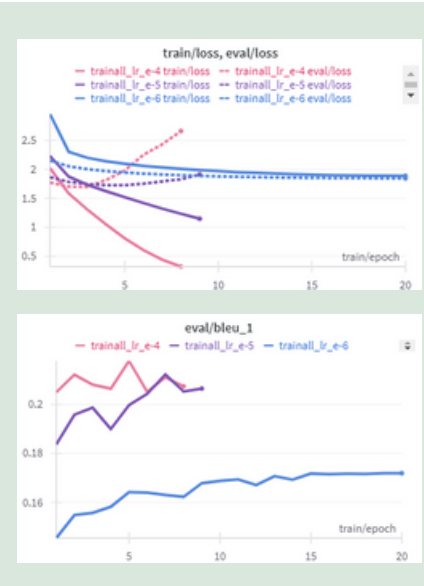
The figure on the left shows the most relevant results from the experiments of freezing layers, one by one, of GPT-2, while keeping ViT all frozen.

We can observe that the best model is the one where the embedding layers and the GPT-2 blocks were frozen up to block number 7. This model exhibited the least overfitting while achieving the best metrics, making it the most effective choice among the tested configurations.

Comparing the best model mentioned before (yellow) and the model with no frozen layers (green), we can observe that the overfitting has been clearly reduced, compensating for the slight reduction in performance (0.01 decay).



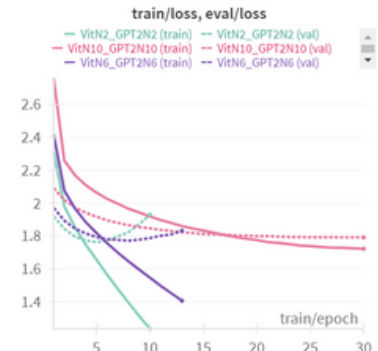
## VIT FINE-TUNE + GPT-2 FINE-TUNE



We first trained the whole encoder and decoder with different learning rates. We tried two approaches one using the same learning rate for both encoder and decoder and the other using a specific learning rate for each. This second approach shows the best results as, even though it has more overfitting, the overall metrics were better, for instance, when comparing the BLEU-1 score, as shown in the image on the right.

$N_{ViT}$	$N_{GPT-2}$
2	2
6	6
7	7
10	10
2	7
7	2
5	7

This experiment consisted on freezing different combinations of layers of ViT and GPT-2 (freeze up until layers  $N_{ViT}$ ,  $N_{GPT-2}$ ). The table on the left states all experiments conducted, and the figure on the right shows the most relevant results.



The models with  $N_{ViT}=N_{GPT-2}=2$  and  $N_{ViT}=N_{GPT-2}=6$  quickly overfit, their training loss drops rapidly while their validation loss starts increasing. In contrast, the model with  $N_{ViT}=N_{GPT-2}=10$  shows a much smaller overfitting gap, with training and validation losses staying closer together. Although both models reach similar validation loss values in the end, the model with  $N_{ViT}=N_{GPT-2}=10$  was considered as the best as it generalizes much better.

**CONCLUSION:** The loss does not always reflect the true quality of the generated captions. Therefore, when loss and evaluation metrics lead to different conclusions, it's better to prioritize the metrics—especially if the goal is high-quality text generation. As a future improvement, early stopping based on BLEU rather than loss (or a combination of both) could better align training with the task objective.

# QUALITATIVE RESULTS

	GroundTruth	Last Week	ViT🔥GPT❄️	ViT❄️GPT🔥	ViT🔥GPT🔥
	Zucchini-Lentil Fritters With Lemony Yogurt	coconut - and -	pucchini andFemoniled-illedter with Carong andurt	Cucchini andandimeil Chickenritters Baconong Butterurt	Cucchini-Stimeil Friedritters Baconony Yogurt
	Dark Chocolate Avocado Brownies	chocolate -	Dark Chocolate Avocado Brownies	Ch Chocolate-ocado andies	Ch Chocolate Brownocado Brownies
	Grilled Chicken with Board Dressing	grille - with -	ailed Chicken with Greeningressing	Grilled Chicken with Lemon-ressing and	Grilled Chicken with Lemon-ressing
	Coconut-Lime Dressing	coconut -	aaulonut andbasedemon-ressing with	Cuconut-Cime Saladressing	Shoconut-Cime Saladressing

## CONCLUSIONS

As we can see from both quantitative and qualitative results, all three methods explored this week achieved better results than the previous week. The best performance was achieved by fine-tuning both ViT and GPT-2. The predictions are much closer to the ground truth than those from last week, with more word roots resembling the original. While some predicted words in the captions do not directly match the ground truth, they are still relevant to the image. For instance:

- In "Grilled Chicken with Board Dressing," the generated caption is "Grilled Chicken with Lemon-ressing." Although the word "lemon" is not in the ground truth, there are lemons visible in the image.
- In "Coconut-Lime Dressing," the predicted caption is "Shoconut-Cime Saladressing." While it slightly differs, for example by adding the word "salad," the image indeed depicts a salad, making the prediction contextually appropriate.

## QUANTITATIVE RESULTS

	Pre-train	Last week	ViT🔥GPT❄️	ViT❄️GPT🔥	ViT🔥GPT🔥
BLEU_1	0.036	0.127	0.155	0.196	0.218
BLEU_2	0.001	0.031	0.058	0.095	0.113
ROUGE_L	0.058	0.106	0.119	0.175	0.196
METEOR	0.026	0.063	0.079	0.111	0.130

🔥 Fine-tune ❄️ Freeze



# QWEN/QWEN2.5-VL-7B-INSTRUCT

	Prompt 1 (baseline)	Prompt 2	Prompt 3 (descriptive +examples)
BLEU_1	0.139	0.237	0.244
BLEU_2	0.064	0.125	0.126
ROUGE_L	0.205	0.228	0.221
METEOR	0.177	0.176	0.180

**Prompt 1:** "Give 1 phrase describing the food in the image from its key ingredients."

**Prompt 2:** "Give a recipe title describing the food in the image. If no title is found give an empty string."

**Prompt 3:** "generate exactly 1 title for the recipe/food/drink based on its key ingredients. Each title should be clear, catchy, and descriptive, highlighting the most important or unique ingredients in the dish. For example, if the key ingredients are potatoes and seasoning, a title might be 'Crispy Salt and Pepper Potatoes'. For a dish with mac and cheese and Thanksgiving flavors, a title could be 'Thanksgiving Mac and Cheese'. You can also include elements that reflect the style or theme of the recipe (e.g., 'Italian Sausage and Bread Stuffing'). Return only the title."



**Model:**  
Qwen 2.5;  
7B  
parameter  
vision  
language  
model  
optimized  
for  
multimodal  
tasks like  
image to  
text  
generation.

We noticed  
that being  
more  
descriptive  
and  
providing  
examples in  
our prompt  
enhanced the  
performance  
on our test  
set

# QUALITATIVE RESULTS



- **Ground truth:** Zucchini-Lentil Fritters With Lemony Yogurt
- **Prompt 1:** Zucchini and Cheese Fritters
- **Prompt 2:** Zucchini Fritters
- **Prompt 3:** Zucchini and Feta Fritters



- **Ground truth:** 3-Ingredient Brown-Butter Shortbread
- **Prompt 1:** The food in the image appears to be a plate of shortbread cookies.
- **Prompt 2:** Shortbread Cookies with Coffee
- **Prompt 3:** Caramelized Rosemary Shortbread Cookies



- **Ground truth:** Anchovy and Rosemary Roasted Lamb
- **Prompt 1:** The image depicts a plate of roasted meat with potatoes and carrots, garnished with herbs.
- **Prompt 2:** Roast Beef with Root Vegetables and Herbs
- **Prompt 3:** Herb-Crusted Roast Beef with Garlic Mashed Potatoes

Prompt 3 provides richer descriptions but occasionally introduces irrelevant details (e.g., "caramelized rosemary" in shortbread).

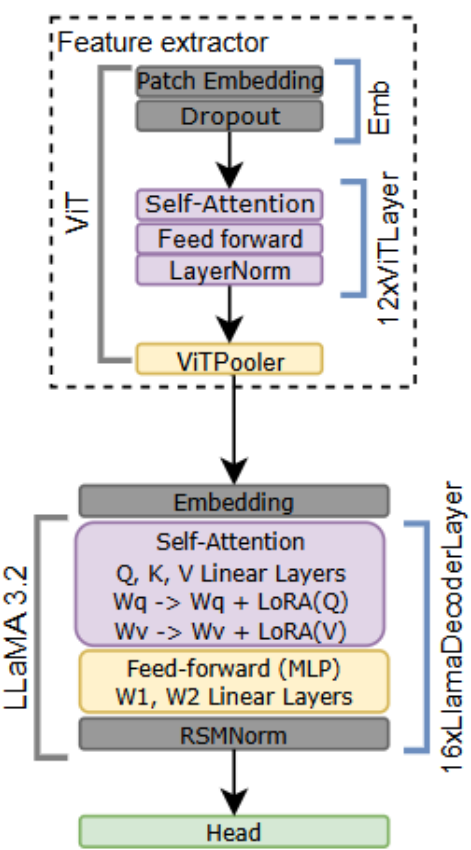
# LLAMA 3.2 + LORA

To enhance the baseline vision-language model, we replaced the decoder with a LLaMA 3.2-1B and a LLaMA 3.2-3B models fine-tuned using Low-Rank Adaptation (LoRA), a lightweight and parameter-efficient method. LoRA allows us to adapt large pre-trained models with significantly fewer trainable parameters by injecting trainable low-rank matrices into attention layers. This setup is particularly valuable when computational resources are limited or when rapid task-specific adaptation is required. We kept the visual backbone (ViT) frozen and used it solely as a feature extractor. This configuration is designed to efficiently handle complex captioning tasks with minimal computational overhead.

## ARCHITECTURE OVERVIEW

### Feature Extractor

In this setup, ViT serves purely as a feature extractor. The output of the final transformer layer is pooled into a single vector that summarizes the visual content. This approach allows for global reasoning over the image without relying on inductive biases inherent to convolutional architectures.



### Decoder

#### Embedding

Visual feature vector is projected through an embedding layer to match the input dimensionality of the language model.

#### 16 x LLaMADecoderLayer

The core of Llama 3.2 consists of 16 LLaMADecoderLayer blocks, each progressively refining the token sequence conditioned on the visual representation. The following architecture is repeated across all 10 layers, progressively enriching the generated sequence with both visual grounding and linguistic coherence.

#### 1. Self-Attention with LoRA

Linear Layers project each token embedding into query (Q), key (K), and value (V) spaces. Lightweight adaptation modules are applied to Q and V to enable efficient fine-tuning with a small number of trainable parameters. This works by adding a low-rank residual to the original weights, which allows the model to adapt to multimodal tasks without retraining the full parameter set.

#### 2. Feed-forward Network (MLP)

Following attention, a gated MLP block refines the token representations. This gating mechanism allows the model to dynamically control information flow through the feed-forward path, improving expressivity without significantly increasing the parameter count.

#### 3. RMSNorm

Applied to stabilize training by scaling activations based on their root-mean-square magnitude.

## TRAINING STRATEGY & EXPERIMENTS

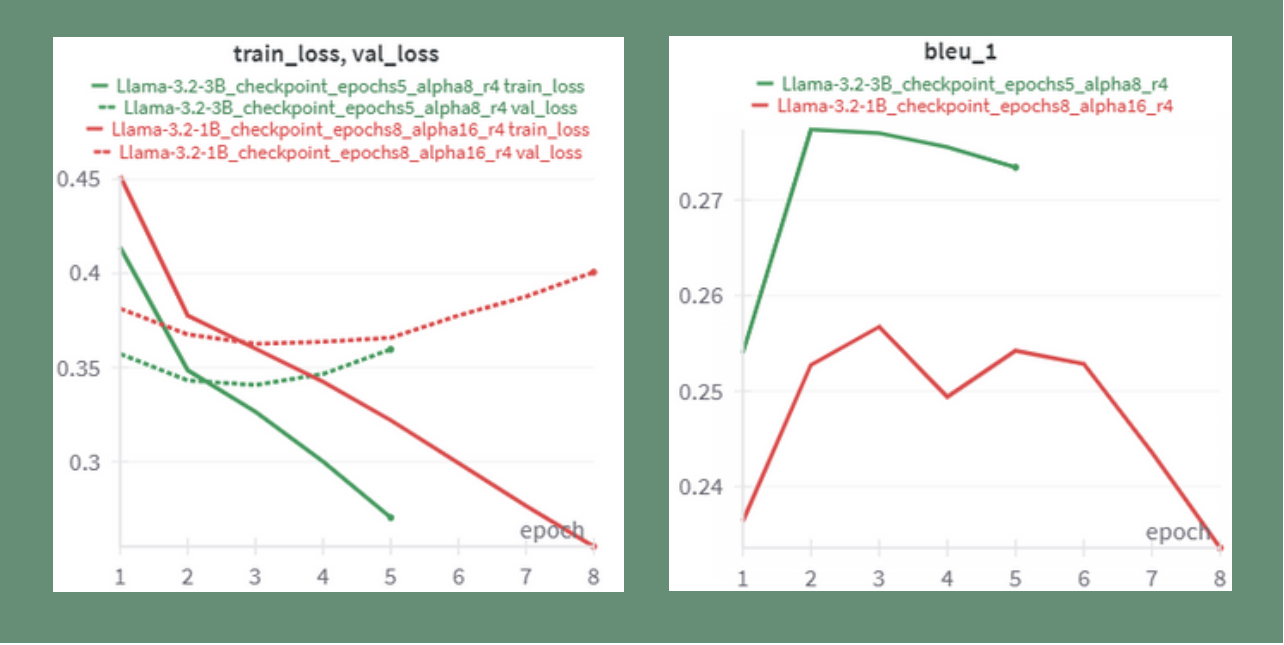
LoRA modifies the original architecture by injecting low-rank matrices into specific linear layers, such as those found in attention and feed-forward networks. These adapters are the only components updated during training, keeping the base model frozen. This allows for fast fine-tuning with a much smaller memory footprint and without compromising on model capacity.

We analyze the impact of different Lora configurations. Specifically, with :

- **r** (rank): that controls the dimensionality of the low-rank matrices used to approximate the original weight updates.
- **alpha**: acts as a scaling factor for the LoRA updates, affecting how strongly the low-rank updates influence the model.

We also tested two different dropout rates (0.05 and 0.1) to evaluate regularization effects, and kept the target modules fixed at ["q\_proj", "v\_proj"], which are key projection layers in transformer architectures where LoRA is applied.

The results displayed here are from experiments using LLaMA 3.2-1B and LLaMA 3.2-3B models. Both models achieved very similar performance. In fact, the difference in BLEU-1 score between the two configurations is as small as 0.03, highlighting that even the smaller LLaMA 1B model can reach competitive performance with proper fine-tuning using LoRA.



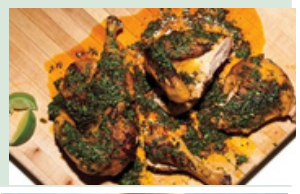



This demonstrates that LoRA can be a powerful and efficient fine-tuning method, enabling lighter models to perform nearly on par with their larger counterparts.



# LLAMA 3.2 + LORA

## QUALITATIVE RESULTS

	Groundtruth	Llama 3.2 1B	Llama 3.2 3B
	Zucchini-Lentil Fritters With Lemony Yogurt	Pucchini andemonil Fritters Lemonemony Yogurt	Grucchini andentil Saladritters with Yogemony Yogurt Sauce
	3-Ingredient Brown-Butter Shortbread	P-Ingredient ChocolateiesButter Chocolatebread	P-Ingredient ChocolateiesButter Shortbread
	Grilled Chicken with Board Dressing	Roilled Chicken with Lemonering	Grilled Chicken with Lemon Buttering
	Coconut-Lime Dressing	Pconut Chickenime Chickening	Spconut Riceime Riceing

## CONCLUSIONS

The 3B model produces captions that are more aligned with the ground truth. The improvements are modest but consistent, indicating better linguistic precision and contextual relevance:

- **BLEU**: Captures more correct n-grams like “Grilled Chicken” and “with Yogurt.” The 3B model preserves more exact word pairs, boosting precision.
- **ROUGE\_L**: Retains longer word sequences in the right order. For example, “P-Ingredient [...] Shortbread” in 3B is closer in structure to the ground truth.
- **METEOR**: Handles stems and semantic similarity—3B’s “Grucchini” ≈ “Zucchini”, “Yogemony” ≈ “Lemony”—which METEOR rewards even if words aren’t exact matches.

## QUANTITATIVE RESULTS

	Llama 3.2 1B	Llama 3.2 3B
BLEU_1	0.260	0.269
BLEU_2	0.134	0.141
ROUGE_L	0.223	0.228
METEOR	0.155	0.160



# SUMMARY

	Groundtruth	ViT🔥 GPT🔥	Qwen*	Llama 3.2 3B
	Zucchini-Lentil Fritters With Lemony Yogurt	Cucchini-Stimeil Friedritters Baconony Yogurt	Zucchini and Feta Fritters	Grucchini andentil Saladritters with Yogemony Yogurt Sauce
	Dark Chocolate Avocado Brownies	Ch Chocolate Brownocado Brownies	Chocolate Avocado Brownies	P Chocolate andocado Mies
	Grilled Chicken with Board Dressing	Grilled Chicken with Lemon-ressing	Herb-Crusted Roasted Chicken with Garlic and Lime	Grilled Chicken with Lemon Buttering
	Coconut-Lime Dressing	Shoconut-Cime Saladressing	Vietnamese Noodle Salad with Shrimp and Tomatoes	Spconut Riceime Riceing

	Last week	ViT🔥	GPT🔥	Qwen*	Llama 3.2 3B
BLEU_1	0.126	0.218	0.244	0.269	
BLEU_2	0.049	0.113	0.126	0.141	
ROUGE_L	0.159	0.196	0.221	0.228	
METEOR	0.062	0.130	0.180	0.160	

🔥 Fine-tune      \*Prompt 3 (descriptive +examples)

## KEY OBSERVATIONS

- The ViT + GPT model, despite being fine-tuned, lagged behind other models' metrics, but is more accurate than last week's results.
- Using prompt-based inference helped to generalize, performed better than the fine-tuned ViT + GPT baseline, highlighting the power of large pretrained models even without fine-tuning.
- LLaMA 3.2 3B outperformed the other models, achieving the highest scores in BLEU-1, BLEU-2 and ROUGE-L.
- Qualitatively, LLaMA 3.2 3B produced more semantically rich and coherent captions, even when wording differed from the ground truth.
- In some cases predicted words in the captions do not directly match the ground truth, but they are still relevant to the image, for instance:
  - In "Grilled Chicken with Board Dressing," the generated caption is "Grilled Chicken with Lemon Buttering." Although the word "lemon" is not in the ground truth, there are lemons visible in the image.