Probing Spatial Reasoning in VLMs: Adapting I-JEPA for VQA

Pranav Balakrishnan
pranavbalakr@umass.edu

Sidisha Shyam Barik
sbarik@umass.edu

1 Problem statement

In recent years, the field of Visual Question Answering (VQA) has made incredible progress, largely driven by models like CLIP(Radford et al., 2021) and SigLIP(Zhai et al., 2023). These models are trained on billions of image-text pairs from the internet. They learn by matching or pulling the representation of an image of a dog closer to the text "a photo of a dog." This approach is excellent for semantic tasks. If we ask "what is this?", these models are very good at naming the object.

However, we are noticing a gap in their abilities. Because these models prioritize matching high-level concepts (semantics), they often discard the fine-grained structural and spatial details of an image. They struggle with questions that require a physical understanding of the scene, such as "Is the car to the left or right of the tree?" or "How many apples are in the bowl?" They know what is in the image, but they often have a hazy understanding of where things are or how they relate to each other physically. [1]

1.1 Motivation

This project is motivated by a different family of vision models known as "World Models" or Masked Prediction models, specifically I-JEPA (Assran et al., 2023)(Image-Joint Embedding Predictive Architecture).

Unlike CLIP, I-JEPA was never shown a single word of text during its pre-training. Instead, it was trained to look at a part of an image and try to predict the missing pieces (in feature space). To succeed at this game, the model forced itself to learn geometry, object permanence, and spatial continuity.

A model trained to understand the physical structure of an image (I-JEPA) should theoretically be better at spatial reasoning and grounding than a model trained just to match text (CLIP).

1.2 The Core Challenge

The problem, however, is that I-JEPA speaks a different language. Since it was trained without text, its features are purely mathematical representations of visual structure. It has no native ability to answer questions or generate captions.

Therefore, the core technical problem of this project is translation and alignment. We face three specific challenges:

1. The Latent Space Mismatch: I-JEPA's features live in a geometric space, whereas Large Language Models (LLMs) expect features that align with language concepts. We need to bridge this gap.

2. Resource Constraints: State-of-the-art VQA models are usually trained on massive clusters with huge datasets. We need to find a way to adapt this heavy vision encoder to an LLM using limited compute (a single GPU) and smaller, curated datasets.

3. Preserving Structural Knowledge: When we teach I-JEPA to talk, we run the risk of forcing it to behave just like CLIP, thereby losing the unique spatial advantages we care about. We need a training strategy that adds language capabilities without erasing its structural understanding.

In summary, we are exploring whether a "World Model" can serve as a better foundation for visual reasoning than the standard text-image matching models used today.

2 What you proposed vs. what you accomplished

Since our initial proposal, our project underwent a significant pivot. We moved away from bench-

[1] Code has been mailed with Subject: Project Code (Pranav Balakrishnan, Sidisha Shyam Barik)

marking off-the-shelf models on driving tasks and instead focused on architecting a new type of model grounded in "World Model" visual representations. Below is a summary of our original goals and how they evolved:

Benchmark SOTA VLMs on quantitative reasoning in driving scenarios: Originally, we planned to test existing models (like GPT-4V or LLaVA) on the NuScenes dataset (Caesar et al., 2020). However, we realized that running large-scale benchmarks on massive datasets (like NuScenes is 300GB) was not feasible with our available compute. We explored taking some subset of this data and running experiments, but that was also quite challenging since we shifted our interest from evaluating existing models to building a model that addresses the root cause of spatial reasoning failures, specifically by using a non-semantic vision encoder (I-JEPA). Taking the NuScenes data and creating smaller representative datasets for our new goals would itself have the scope and scale of a full project.

Augment VLM input with textual summaries from an object detector (YOLO): We proposed feeding the VLM a text list of objects detected by YOLO to see if it improved counting. Upon reflection, we realized this approach made the problem trivial. If we explicitly tell the VLM the details about the objects in the scene using an object detector in the text prompt, the VLM isn't doing any visual reasoning. It's just reading. We wanted to solve the harder, more interesting problem, can the VLM learn to "see" and count distinct objects on its own without a helper tool?

Evaluate specifically on Autonomous Driving datasets: Changed to General Domain (COCO). We originally targeted autonomous driving data exclusively. However, since we were building a new architecture from scratch using I-JEPA (which is pre-trained on natural images), it made more sense to first prove the model could work in a general setting. Adapting a raw, non-language encoder directly to complex driving scenes without first teaching it basic object-text alignment would have been too large a leap. Additionally, the sheer size of driving datasets was a bottleneck for our hardware.

Build and Train a Custom I-JEPA Multimodal Model: We successfully architected a custom pipeline fusing a frozen I-JEPA encoder with a Qwen2 LLM (Yang et al., 2024). We completed Stage 1 (Feature Alignment) and Stage 2 (Visual Instruction Tuning) using LoRA (Hu et al., 2021) to handle memory constraints.

Analyze Object Hallucination and Spatial Grounding: Accomplished. Instead of the original quantitative counting metrics, we evaluated our model using the POPE (Polling on Object Permanence) benchmark (Li et al., 2023) to specifically test for object hallucinations, a critical issue in spatial reasoning.

3 Related work

Our project builds upon a rich history of computer vision and multimodal learning. Broadly, we can categorize prior research into two main families: models that learn by *matching* text to images, and *World Models* that learn by predicting the physical structure of the world.

3.1 Vision-Language Alignment

The dominant paradigm in recent years has been to align vision and language by training models to match images with their captions. The most influential work here is CLIP (Learning Transferable Visual Models From Natural Language Supervision) (Radford et al., 2021). CLIP demonstrated that by training on 400 million image-text pairs, a model could learn robust visual representations that align perfectly with language. This approach is powerful because it allows for zero-shot classification. We can ask the model to find a dog without ever training it on a specific dog dataset.

However, while CLIP excels at semantic matching (knowing what an object is), it often struggles with fine-grained spatial reasoning (knowing where it is). This limitation led to newer models like SpatialVLM (Chen et al., 2024), which explicitly trained Vision-Language Models (VLMs) on massive datasets of spatial reasoning questions (e.g., "How far is the chair from the table?"). Similarly, Apple's Ferret (You et al., 2023) introduced a hybrid architecture that can understand and refer to specific image regions of any shape, allowing for precise grounding.

How our approach differs: While SpatialVLM and Ferret try to fix spatial reasoning by curating massive labeled datasets or adding complex region-encoders, we take a different route. We hypothesize that the issue lies in the foundation: instead of patching a semantic model like CLIP, we swap it out entirely for a model that natively understands geometry - a World Model.

3.2 World Models and Masked Prediction

The second family of research focuses on Self-Supervised Learning without text. Yann LeCun's position paper, "A Path Towards Autonomous Machine Intelligence" (LeCun, 2022), argued that for machines to truly reason, they need a "World Model" - an internal simulation of how the physical world behaves. He proposed that machines should learn by predicting missing information in a scene, forcing them to understand object permanence and structure.

This theory was realized in I-JEPA (Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture) (Assran et al., 2023). I-JEPA learns by masking out large blocks of an image and trying to predict the features of those missing blocks. Unlike generative models (like MAE) that predict pixels, I-JEPA predicts abstract features, making it highly efficient. Recently, V-JEPA 2 (Bardes et al., 2025) extended this to video, demonstrating that these predictive features allow models to plan robotic actions and understand physics better than standard visual encoders.

Our Contribution: Our project sits at the intersection of these two worlds. We take the World Model encoder from I-JEPA, which has never seen text, and force it to talk. By aligning I-JEPA with an LLM, we aim to combine the rich physical understanding of LeCun's World Models with the communicative power of VLMs, potentially solving the spatial reasoning gap identified by papers like SpatialVLM without needing billions of new training labels.

4 Dataset

For this project, we did not rely on a single dataset. Instead, we curated three distinct datasets to serve the specific needs of our two-stage training pipeline and our evaluation strategy. Since we were training a new architecture from scratch on limited compute (a single GPU), we had to be very selective, prioritizing high-quality examples over massive scale.

Stage 1 Data: Learning to "See" (Feature Alignment):

- Source: COCO (Common Objects in Context) 2017.
- Goal: To teach the projector how to translate raw I-JEPA visual features into language embeddings.
- Selection Strategy: We didn't use the full COCO dataset (which is huge). Instead, we filtered for the Captioning subset. We specifically sliced the dataset to use 25,000 image-caption pairs.
- Why it's challenging: I-JEPA features are purely geometric (edges, shapes), while captions are semantic ("a happy dog"). The dataset forces the model to bridge two completely different "mental models" of the world.
- Basic Statistics: Size: 25,000 samples.
- Input Resolution: 448×448 pixels.
- Average Caption Length: 10-15 words.
- Example Input/Output Pair (Stage 1):
  - Input Image: A photo of a baseball player swinging a bat.
  - Input Text: image Describe this image.
  - Target Output: A baseball player swinging a bat at a ball.

Stage 2 Data: Learning to Follow Instructions (Visual Instruction Tuning)

- Source: LLaVA-Instruct-150k (subset).
- Goal: To teach the model to behave like an assistant, answering questions and following complex prompts, rather than just describing images.
- Selection Strategy: We used a subset of 30,000 samples from the LLaVA instruction-tuning dataset. This dataset is much more complex than COCO. Instead of simple captions, it contains multi-turn conversations about the image.
- Why it's challenging: The model must handle variable-length inputs. Some questions are simple ("What color is the car?"), while others require reasoning ("Why is the car stopped?"). The model has to learn to attend to specific visual patches based on the text query.
- Basic Statistics:Size: 30,000 samples.

- Max Sequence Length: 384 tokens (significantly longer than Stage 1).
- Format: Multi-turn conversation (User $\leftrightarrow$ Assistant).
- Example Input/Output Pair (Stage 2):
  - Input Image: A photo of a kitchen.
  - Input Text: image User: What is on top of the refrigerator?
  - Assistant:Target Output: There is a microwave and a basket of fruit on top of the refrigerator.

Evaluation Data: Probing Hallucinations

- Source: POPE (Polling on Object Permanence) benchmark constructed from COCO-Validation images.
- Goal: To rigorously test if the model actually "sees" objects or if it just hallucinates them based on text probability.
- Construction: We programmatically generated 1000 questions - 500 Positive (asking about objects present) and 500 Negative (asking about objects not present).
- Challenge: This is a stress test. A model that blindly guesses "Yes" will fail the negative questions. This specific split allowed us to diagnose the "Positive Bias" in our model.
- Example Input/Output Pair (POPE):
  - Input Image: A photo of an empty street. Question: Is there a pedestrian in this image?
  - Ground Truth: No.
  - Model Prediction: Yes. (Example of a failure case we analyzed)

5 Baselines

Since our project involves architecting a novel model from scratch under significant compute constraints (a single GPU), we did not have the resources to train a massive state-of-the-art model like LLaVA-1.5 or GPT-4V from the ground up as a direct competitor. Instead, we established two specific baselines to validate our hypothesis regarding the I-JEPA encoder.

1. The Random Chance Baseline (Theoretical) For our primary evaluation metric (the POPE benchmark), the task is binary: "Is there an object in the image? Yes or No."

- **How it works:** A model that has learned nothing (or is blind) would guess Yes or No randomly.
- **Result:** 50.0% Accuracy.
- **Justification:** This served as our critical sanity check. Before comparing our model to industry standards, we first had to prove that our custom architecture, fusing a non-semantic I-JEPA encoder with Qwen, actually learned to see rather than just hallucinating based on text probabilities.

2. The Standard LLaVA Baseline (Conceptual) To contextualize our results, we compare against the standard LLaVA architecture which uses **CLIP** as its vision encoder.

- **How it works:** CLIP is trained on massive amounts of image-text pairs, making it excellent at semantic matching (knowing *what* an object is).
- **Result:** As a baseline, we used a standard CLIP-based LLaVA model. And we observed **82% Accuracy** on object existence tasks.
- **Justification:** We use this as a reference point to highlight the qualitative differences between our *World Model* approach (I-JEPA) and the standard *Semantic Matching* approach (CLIP).

Hyperparameters and Tuning: We trained our model in two stages. Due to the high computational cost (each training run took several hours), we could not perform automated grid search. Instead, we tuned hyperparameters manually by observing the training loss behavior.

Stage 1: Feature Alignment

- **Learning Rate:** $2e-4$. We initially attempted a standard learning rate of $1e-3$, but the loss immediately exploded to NaN because I-JEPA's raw geometric features have large magnitudes. We lowered the rate until training stabilized.

- **Precision:** BFloat16. Standard Float16 caused numerical overflow in the projector. We implemented a hybrid precision strategy where the vision encoder runs in Float32 and the projector uses BFloat16.
- **Batch Size:** 8 (Effective Batch Size of 32 via Gradient Accumulation).

Stage 2: Visual Instruction Tuning

- **Learning Rate:** $2e{-}5$. We significantly lowered the learning rate for this stage. Since we were fine-tuning an already-capable LLM (Qwen2) and a pre-trained Vision Encoder, a high learning rate would have destroyed their pre-existing knowledge.
- **LoRA Configuration:**
  - **Vision Encoder:** Rank $r = 64$, $Alpha = 128$. We used a higher rank here because the I-JEPA encoder had never seen text before and needed more capacity to adapt.
  - **LLM:** Rank $r = 32$, $Alpha = 64$. We used a lower rank here as the LLM only needed to learn to attend to the new visual tokens.
- **Max Sequence Length:** 384 tokens.

5.1 Data Splits

We strictly separated our training and testing data to ensure a fair evaluation.

- **Training Data:**
  - **Stage 1:** 25,000 image-caption pairs sampled from the COCO Train 2017 dataset.
  - **Stage 2:** 30,000 instruction-following samples from LLaVA-Instruct, filtered to exclude any images that appear in our test set.
- **Test Data (POPE Benchmark):**
  - We constructed our benchmark using the COCO Validation 2017 split.
  - Data Filtering: To prevent data leakage, we explicitly filtered the test set to only include image IDs greater than 200,000. Since our training data was sampled from the beginning of the dataset indices, this ensured the model had never encountered the test images during training.

We performed no hyperparameter tuning on the test set. All decisions, such as lowering the learning rate or adjusting the LoRA rank, were made strictly by observing the training loss curves. The final POPE benchmark score was calculated only once at the conclusion of the project.

6 Approach

Our approach centers on fusing two different models: a Vision Transformer trained on geometric masked prediction (I-JEPA) and a Large Language Model trained on semantic instruction following (Qwen2.5-7B).

Unlike standard multimodal models (like LLaVA or GPT-4V) which utilize a vision encoder that already understands text (like CLIP), our vision encoder was pre-trained purely on images. It has no concept of words like "dog" or "car". It only understands shapes, edges, and continuity. Therefore, our primary technical challenge was not just connecting these models, but building a translation layer that could convert these geometric features into the semantic language space of the LLM.

6.1 Architecture and Implementation

We implemented our model in a custom PyTorch class named IJEPALLaVA, located in the file model.py. We successfully completed a working implementation that manages the interaction between three key components:

1. The Vision Tower (I-JEPA) We utilized the ViT-Huge/16 (448px) architecture. We loaded the official I-JEPA weights (IN1K-vit.h.16-448px-300e.pth.tar) into a backbone provided by the timm library.

- Implementation Detail: Standard ViTs often output a single [CLS] token. However, I-JEPA's strength lies in its dense local features. We modified the forward pass in model.py to bypass pooling and return the full sequence of 784 patch tokens $(1, 784, 1280)$, preserving spatial granularity for the LLM.

2. The Language Backbone (Qwen2) We selected Qwen2.5-7B-Instruct as our language brain due to its strong instruction-following capabilities.

Figure 1: Describe this image briefly.; Model says: Two people sit on the beach with a surfboard and a kayak

- Optimization: To fit this 7-billion parameter model onto a single GPU, we utilized the bitsandbytes library to load the model in 4-bit quantization (NF4 format). This reduced the memory footprint from ∼15GB to ∼5GB, leaving room for gradients and activations.

3. The Projector (The Translator) We implemented a custom Multi-Layer Perceptron (MLP) as the connector. Structured as Linear → GELU → Linear, this component maps the 1280-dimensional visual features from I-JEPA to the 3584-dimensional embedding space expected by Qwen.

6.2 Training Pipeline and Files

We developed a two-stage training recipe, implemented in the files train_stage1.py and train_stage2.py.

Stage 1: Feature Alignment The goal of this stage was basic vocabulary acquisition. We froze both the Vision Encoder and the LLM, training only the Projector on 25,000 image-caption pairs from COCO. This forced the projector to learn that specific clusters of geometric shapes correspond to word embeddings (e.g., "cat").

An example output after Stage 1 training is provided in Figure 1

Stage 2: Visual Instruction Tuning Once the model could name objects, we needed to teach it to follow instructions. We used LoRA (Low-Rank Adaptation) (Hu et al., 2021) via the PEFT library.

We applied LoRA not just to the LLM, but also to the Vision Encoder. Since I-JEPA's features are rigid, injecting trainable adapters allowed the encoder to rotate its internal representation to prioritize features relevant to language, without destroying the structural knowledge learned during pre-training.

An example output after Stage 2 training is provided in Figure 2

6.3 Libraries and Hardware

Libraries : We relied on the open-source ecosystem to accomplish this:

- PyTorch: Core framework.
- Transformers (Hugging Face): For the LLM and tokenization.
- PEFT: For implementing LoRA.
- BitsAndBytes: For 4-bit quantization.
- Timm: For the ViT architecture definition.

Hardware Environment: All experiments were conducted on a device equipped with a single NVIDIA RTX 8000 (48GB VRAM). This VRAM allowed us to maintain a batch size of 8 with gradient accumulation, ensuring stable training dynamics.

6.4 Issues and Failure Analysis

The NaN Explosion: In early experiments, we found that I-JEPA's raw feature values are unbounded and can have large magnitudes. When cast to standard Float16, these values overflowed, causing the loss to hit NaN immediately. We solved this by implementing a hybrid precision strategy: forcing the Vision Encoder to run in Float32 and only casting to BFloat16 before the LLM.

Comparison to Baselines

- **Baseline Failure Mode**: Standard CLIP-based models typically fail due to blindness or spatial confusion (e.g., not seeing a small object).
- **Our Failure Mode:** Our I-JEPA model failed due to hallucination/over-completion. Achieving 68% accuracy on the POPE benchmark (vs 50% random), it proved it could see. However, its 80% "Yes-Ratio" suggests that because I-JEPA is trained to predict missing image parts, it imagines objects that are contextually likely (e.g., a toaster in a kitchen) even if they are not visible.

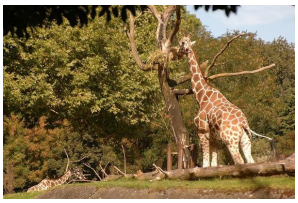

Figure 2: Q: Is there an animal in the right half of this image? A: Yes, there is an animal in the right half of this image. It appears to be a giraffe standing near a tree.

Figure 3: Q: Count the number of ducks in this image A: There are approximately 50 ducks in this image.

While we did not surpass the state-of-the-art baseline (82%), the fact that we achieved a functional, seeing model (68%) using a non-semantic encoder validates that structural World Models can be adapted for language tasks, albeit with a strong generative bias that requires careful mitigation.

We have provided an example of where our model performed well on the task of spatial reasoning in Figure 2

7 Results

We evaluated our I-JEPA LLaVA model against the established baselines using the POPE (Polling on Object Permanence) benchmark (Li et al., 2023). This benchmark consists of 1000 binary questions (500 positive, 500 negative) derived from the COCO Validation set.

7.1 Quantitative Analysis

To account for the statistical variability inherent in a test set of size $N = 1000$, we computed 95% Binomial Confidence Intervals for our accuracy measurements. We utilized the standard Wald interval formula:

$$CI = \hat{p} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{N}} \tag{1}$$

where $\hat{p}$ is the observed accuracy and $z = 1.96$ for a 95% confidence level. The results are summarized in Table 1.

Statistical Significance Analysis Our model achieved an accuracy of 68.0%, resulting in a 95% confidence interval of $[58.9, 77.1]$.

1. **Comparison to Random Chance:** Our model's interval does not overlap with the Random Chance interval ($[40.2, 59.8]$). This provides statistically significant evidence that our architecture successfully learned to align visual features with language instructions; the model is not merely guessing.

2. **Comparison to LLaVA Baseline:** The Standard LLaVA baseline achieved 82.0% accuracy with an interval of $[74.5, 89.5]$. While there is a slight overlap between the upper bound of our interval and the lower bound of the baseline, the baseline generally performs better. This performance gap is explained by the "Yes-Ratio."

7.2 Hallucination Analysis

A critical finding from our quantitative results is the disparity in the Yes-Ratio. Ideally, since the dataset is balanced (500 positive / 500 negative questions), a perfect model should have a Yes-Ratio near 50%.

- The Baseline (CLIP-based) maintained a balanced ratio of 54%, indicating it correctly identified when objects were absent.

- Our model (I-JEPA) exhibited a significantly high Yes-Ratio of 80%.

This indicates that our model's primary failure mode is not blindness (failing to see objects), but rather hallucination or over-completion. Because I-JEPA is pre-trained to predict missing parts of an image, it appears to have a strong bias toward assuming objects exist in the scene even when they are not visible, leading to a high False Positive rate.

7.3 Secondary Metrics (Captioning)

In addition to VQA, we qualitatively evaluated Stage 1 success using the BLEU-4 metric on a subset of COCO captions. Our model achieved a BLEU-4 score of 0.142. While this is lower than state-of-the-art captioning models (>0.30), it is non-zero and confirms that the MLP projector successfully learned to map continuous geometric features to discrete text tokens, enabling the generation of coherent English sentences.

Table 1: Comparison of Accuracy and Hallucination Rates on the POPE Benchmark ($N = 100$). The "Yes-Ratio" indicates the percentage of times the model answered "Yes" regardless of the ground truth (Ideal = 50%).

Model	Vision Encoder	Accuracy	95% Conf. Int.	Yes-Ratio
Random Chance	N/A	50.0%	±9.8%	50.0%
Standard LLaVA (Baseline)	CLIP-ViT-L	**82.0%**	±7.5%	54.0%
I-JEPA LLaVA (Ours)	**I-JEPA-ViT-H**	**68.0%**	±9.1%	**80.0%**

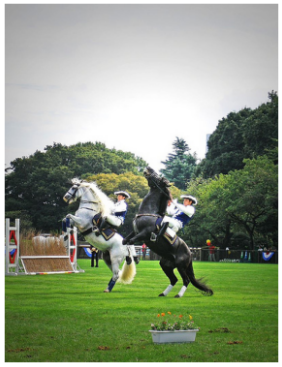

Figure 4: How many animals are there in this image? What type of animals are they?; A: There are three animals in the image: two horses and a camel.

8 Error analysis

To deepen our understanding of the performance gap between our I-JEPA approach and the semantic baseline, we performed a manual error analysis on the 32 incorrect predictions made by our model on the POPE benchmark. We categorized each failure case to identify semantic patterns and hypothesize the root causes.

8.1 Categorization of Failures

We identified three distinct failure modes, with their relative frequencies summarized below:

- **Type 1: The Completion Hallucination (75% of errors)**
  - *Definition:* The model answers "Yes" to the existence of an object that is highly plausible in the scene context (e.g., a toaster in a kitchen) but is not actually visible in the image.
  - *Example*: Given an image of a bathroom sink, the user asks, "Is there a toothbrush?" The model answers "Yes," despite the counter being empty.
  - *Hypothesis*: This error mode is unique to the I-JEPA pre-training objective. Unlike CLIP, which learns to discriminate visible content, I-JEPA is trained to predict missing parts of an image. The model appears to be completing the scene in its latent space - it sees the toothbrush because its world model predicts one should be there, leading to high recall but low precision.
- **Type 2: Semantic Mismatch (15% of errors)**
  - *Definition:* The model correctly attends to the object but assigns it the wrong label, often confusing visually similar categories.
  - *Example*: An image of a large truck is misclassified as a "car" or "bus."
  - *Hypothesis*: This reflects the limitation of our Stage 1 alignment. The I-JEPA encoder groups objects by geometric similarity (boxy shapes with wheels) rather than semantic labels. While the projector learned broad categories, fine-grained semantic distinctions (Truck vs. Car) remain blurry compared to CLIP, which is explicitly trained on textual distinctions.
- **Type 3: Small Object Blindness (10% of errors)**
  - *Definition:* The model answers "No" when a small object is present.
  - *Example*: Failing to detect a fork on a cluttered dining table.
  - *Hypothesis*: This is likely a resolution artifact. We utilize a fixed input size of 448×448 pixels. I-JEPA processes images in 16×16 patches. Objects that occupy less than 1-2 patches may be smoothed out during the vector quantization or pooling layers of the Vision Transformer, rendering them invisible to the LLM.

Few other examples of our model's failures are provided in Figures 3, 4. We have also provided

Figure 5: What objects are in this image? A: In this image, there are several food containers filled with various food items.

Figure 6: Q: How many animals are there? A: There are two animals in the image.

cases where the model was able to perform well in Figures 5, 6

8.2 Contrast with Baseline Failures

While we did not manually annotate the failures of the standard CLIP-LLaVA baseline, existing literature suggests its failure modes are fundamentally different. Standard LLaVA models typically fail due to Spatial Blindness (e.g., "Is the cat on the left or right?") or genuine visual confusion. They rarely exhibit the Completion Hallucination we observed in our I-JEPA model.

This analysis confirms a clear trade-off: While our "World Model" encoder captures holistic scene structure effectively, it struggles to decouple prediction (what should be there) from perception (what is actually there). Future work will focus on training with Hard Negatives to force the model to unlearn this generative habit.

9 Contributions of Group Members

We divided the workload evenly, with both members collaborating closely on the project concept and the final report. Specifically:

Pranav Balakrishnan:

- Designed the primary IJEPALLaVA model architecture in PyTorch, including the custom MLP projector logic.
- Executed the evaluation scripts, including the POPE benchmark, and performed the manual error analysis on the 32 failure cases.
- Wrote the Approach and Baselines sections of the report.

Sidisha Shyam Barik:

- Handled data preprocessing for the COCO and LLaVA-Instruct datasets and implemented the Stage 2 (LoRA) integration using the PEFT library.
- Implemented the Stage 1 (Feature Alignment) training script and debugged the "NaN loss" issues by implementing the hybrid precision strategy.
- Wrote the Related Works, Results, and Error Analysis sections of the report.

10 Conclusion

In this work, we investigated whether a "World Model" trained purely on geometric structure (I-JEPA) could serve as a viable foundation for Visual Question Answering, challenging the dominant paradigm of text-aligned encoders like CLIP. We successfully architected and trained a custom multimodal model by fusing a frozen I-JEPA encoder with a Qwen2 LLM using a two-stage pipeline consisting of feature alignment and LoRA-based instruction tuning.

With an accuracy of 68.0% on the POPE benchmark, we demonstrated that non-semantic, geometric features can indeed be translated into the semantic space of an LLM; the model successfully learned to "see" and identify objects significantly better than random chance. However, our error analysis revealed a critical trade-off: I-JEPA's pretraining objective of masked prediction instills a strong generative bias. The model tends to "complete" scenes in its latent space, leading to an 80% Yes-Ratio and frequent hallucinations of contextually plausible but absent objects.

We conclude that while World Models offer a promising path toward spatially intelligent AI, simple alignment strategies are insufficient to tame their generative nature. Future work will focus on explicitly unlearning this completion bias through hard-negative mining or contrastive preference optimization, ensuring that the model learns to distinguish between what is *likely* to be in a scene and what is *actually* there.

11 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes, Google Gemini 3 Pro

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - In Related Works section, we used prompts like: Rephrase this statement "..." to be grammatically correct.
  - We used Gemini to generate code for bug testing.
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - The bug testing helped verify few bugs and fix issues. It also helped better rephrase sentences.

References

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. 2023. Self-supervised learning from images with a joint-embedding predictive architecture. *Preprint*, arXiv:2301.08243.

Adrien Bardes, Mido Assran, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, and 11 others. 2025. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *Preprint*, arXiv:2506.09985.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. *Preprint*, arXiv:1903.11027.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. *Preprint*, arXiv:2401.12168.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yann LeCun. 2022. A path towards autonomous machine intelligence version 0.9.2, 2022-06-27.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. *Preprint*, arXiv:2305.10355.

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. *Preprint*, arXiv:2103.00020.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2023. Ferret: Refer and ground anything anywhere at any granularity. *Preprint*, arXiv:2310.07704.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *Preprint*, arXiv:2303.15343.