

Appendix

The appendix is organized as follows. First, we show the details of mask-label annotation pipeline and the the patch-aligned dataset format in Appendix A. Then, we introduce the details about experiment setting in Appendix B. Finally, we present visualizations related to token-level alignment—specifically patch-level localization in Appendix C and multi-semantic alignment in Appendix D.

A Patch Aligned Dataset

In this section, we present the format of the Patch Aligned Dataset(PAD) in comparison to the LLaVA pretraining dataset following the mask-label annotation pipeline.

Mask-Label Annotation Pipeline. To address the lack of patch-level annotated data, we develop an automated annotation pipeline for generating the Patch-Aligned Dataset (PAD), designed to refine the LLaVA-pretrained dataset by incorporating fine-grained details. PAD enriches this dataset with detailed annotations, including object tags, bounding box locations, and segmentation masks for individual objects. By incorporating dense, pixel-level grounding information, PAD is designed to enhance fine-grained image-text alignment during the pretraining stage, thereby improving the model’s ability to understand localized regions within the image.

As illustrated in Figure 2, our automated annotation pipeline consists of diverse state-of-the-art models, including Recognize Anything Model (RAM) [23], Grounding DINO [24], and Segment Anything Model (SAM) [25]—to perform grounded image segmentation and object recognition. First, RAM generates object tags from the input image. These tags are then passed to Grounding DINO, which generates bounding boxes for each identified object. Afterward, a Non-Maximum Suppression (NMS) process is applied to filter overlapping bounding boxes based on Intersection over Union (IoU) thresholds. The remaining bounding boxes are passed to SAM, which generates segmentation masks for each object. The pipeline outputs the segmented image with bounding boxes, along with metadata in JSON format, including object tags, bounding box coordinates, and RLE-encoded masks for further analysis.

As shown in Table 9, we annotate the images of the LLaVA pretraining dataset with additional object tags, bounding box coordinates, and RLE-encoded masks stored in a JSON file. The RLE-encoded masks can be decoded back into binary masks that have the same size as the image.

B Experiment Setup

- **Architecture** To evaluate the effectiveness of our method, we ensure a fair comparison by following the same architecture as LLaVA 1.5. Specifically, we use CLIP-ViT-L@336px [33] as the vision encoder \mathcal{E} , Vicuna-1.5-7B[47] as the LLM \mathcal{L} , and a 2-layer MLP as the projector \mathcal{P} . The parameter β follows a linear schedule, increasing from 0 to 5.
- **Training Details** Following the standard training paradigm in LLaVA [1], our training pipeline consists of two stages. In stage 1, keeping the vision encoder \mathcal{E} and LLM \mathcal{L} frozen, we train only the projector \mathcal{P} using our proposed *Patch Aligned Training* method to obtain the patch-aligned projector $\mathcal{P}_{\text{Patch Aligned}}$. In stage 2, we perform supervised fine-tuning on both the LLM \mathcal{L} and the patch-aligned projector $\mathcal{P}_{\text{Patch Aligned}}$. Following LLaVA’s hyperparameters, we optimize all models for 1 epoch using the AdamW optimizer with a cosine learning schedule. The learning rates are set to 1e-3 for pretraining and 2e-5 for instruction tuning. Pretraining requires approximately 8 hours using 8xA5000 GPUs (24G), while visual instruction tuning takes about 10 hours for LLaVA-v1.5-7B on 8xH100 (80G).
- **Dataset** For pretraining dataset, utilizing our automated annotation pipeline, we annotate the 558K subset of the LAION-CC-SBU dataset, which is used as the pretraining dataset of LLaVA. The resulting dataset comprises 2.3M regions, each associated with a segmentation mask, and includes 33.5K unique tags. For fair comparison, we use the same vision instruction tuning dataset as the one in the LLaVA-1.5, containing LLaVA-Instruct [1], TextVQA [48], GQA [49], OCR-VQA [50], and Visual Genome[51].

Table 9: Comparison of LLaVA Pretraining Dataset and Patch Aligned Dataset (Ours).

		
LLaVA Pretraining Dataset	<pre> "image_id" : "00000/00000030.jpg" "size": [448, 336] "caption": "a canyon wall reflects the water on a sunny day in utah." </pre>	
Patch Aligned Dataset (ours)	<pre> "image_id" : "00000/00000030.jpg" "size": [448, 336] "caption": "a canyon wall reflects the water on a sunny day in utah." "labels": [{ "tag": "water", "bbox": [-0.0003204345703125, 182.57894897460938, 447.99951171875, 335.67926025390625], "rle_mask": "k5d4L5000000100000001000100000000000000..." }, { "tag": "cliff", "bbox": [-0.064117431640625, 0.34404754638671875, 447.9346005859375, 182.572509765625], "rle_mask": "S32.:0eE0V:5004LXY2:[fM302M20200N2N3N1010101N20101N20..." }] </pre>	

549 C Patch-Level Localization: More Visualizations

550 Following the micro-scale analysis on patch-level localization in Section 3.2.1, we provide more
 551 examples comparing the ground truth mask M_{GT} and predicted mask M_{pred} generated by three pro-
 552 jectors: the random projector \mathcal{P}_{Random} , pretrained LLaVA 1.5 projector \mathcal{P}_{LLaVA} , and our PatchAligned
 553 Projector $\mathcal{P}_{Patch\ Aligned}$. As shown in Figure 4, the PatchAligned Projector predicts more accurate
 554 masks.

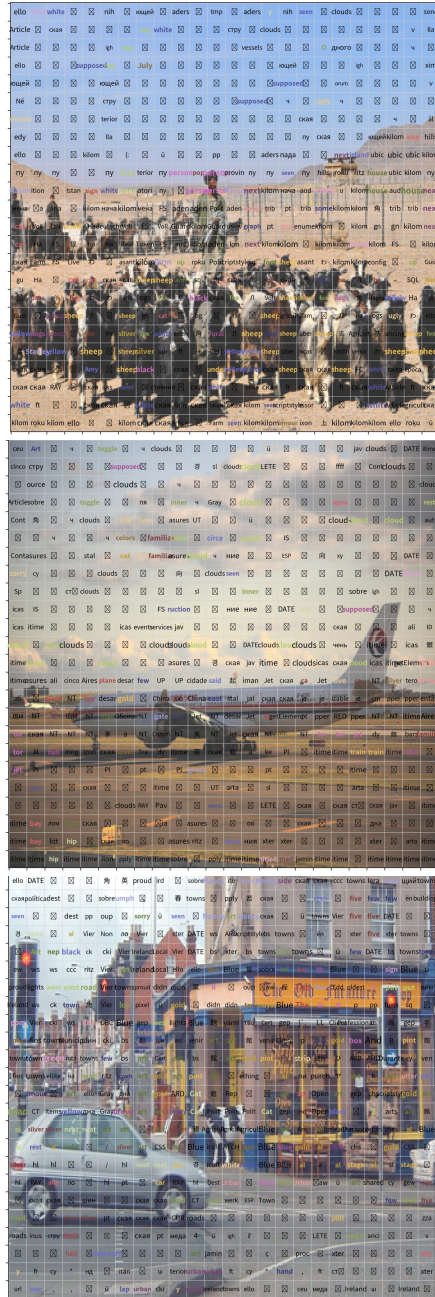


Figure 4: Additional visualization for patch-level localization.

555 D Multi-Semantic Alignment: More Visualizations

556 We begin by showing the first iteration of matching pursuit, which finds the token in the LLM
 557 embedding space that has the highest similarity with the vision embedding. We show the full
 558 tokenmap in Figure 5, displaying the found token for each vision patch. We use font size to represent
 559 similarity. Tokens recognizable by NLTK are shown in color, while unrecognized tokens remain
 560 black. For LLaVA, only partial areas or objects achieve alignment. In contrast, PatchAligned LLaVA
 561 achieves better alignment across most patches.

LLaVA



PatchAligned LLaVA



Figure 5: Additional visualization for tokenmap comparing LLaVA and PatchAligned LLaVA.

562 Next, following Section 3.2.2, we apply matching pursuit on PatchAligned LLaVA for 5 iterations.
 563 As shown in Figure 6, the semantic meanings are decoded for each iteration, with cosine similarity
 564 decreasing across iterations.

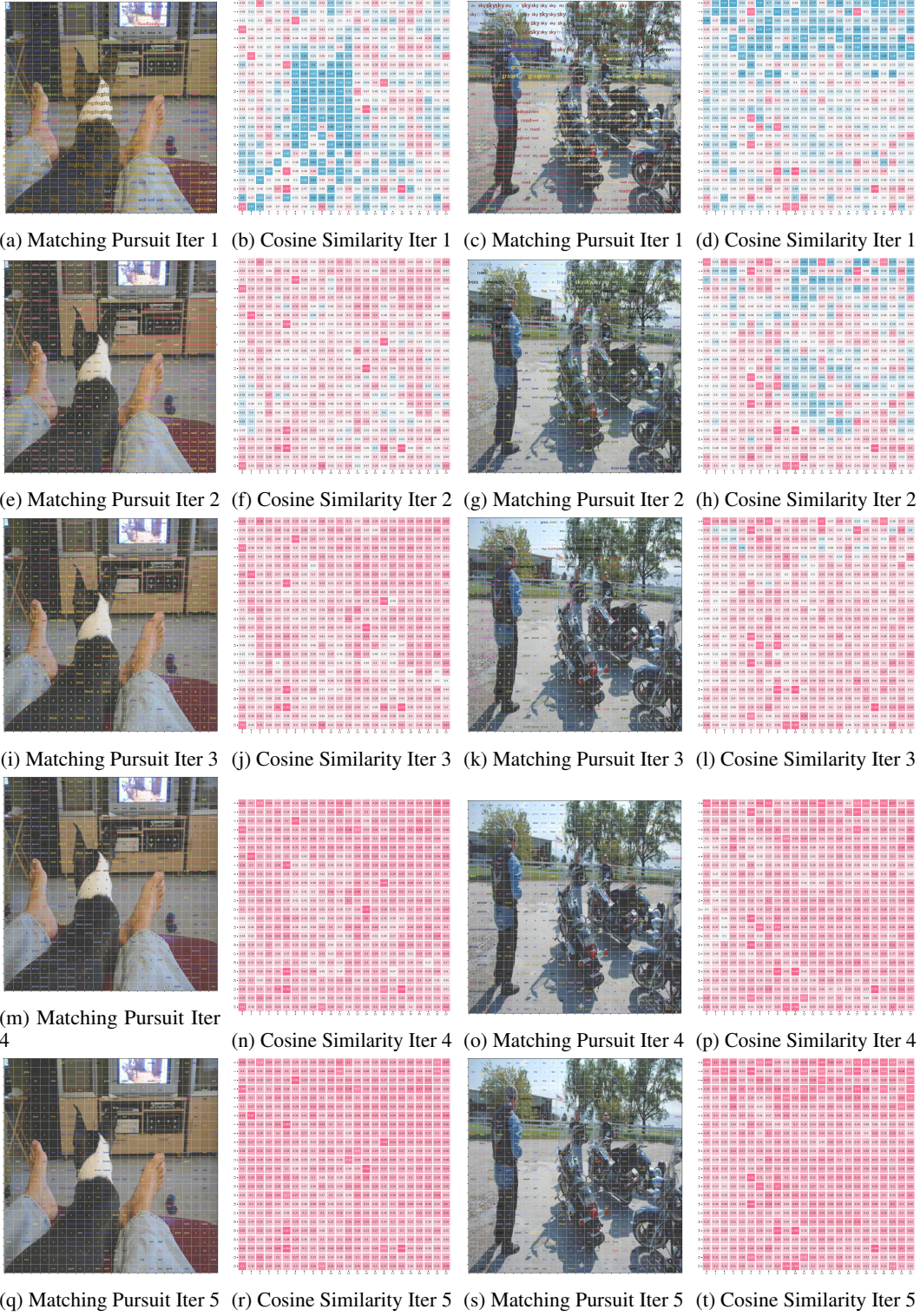


Figure 6: Perform Matching Pursuit using PatchAligned LLaVA. Each row represents an iteration, with selected tokenmap (Column 1,3) and cosine similarity maps (Column 2,4).