SPIN: Hierarchical Segmentation with Subpart Granularity in Natural Images - Supplementary Materials

Josh Myers-Dean[1], Jarek Reynolds[1], Brian Price[2], Yifei Fan[2], and Danna Gurari[1,3]

[1] University of Colorado Boulder, [2] Adobe, [3] University of Texas at Austin

1 Supplementary Materials

This document supplements the main paper with the following:

1. SPIN dataset creation. (supplements **Section 3.1**)
2. Crowdsourcing implementation. (supplements **Section 3.1)**
3. SPIN analysis. (supplements **Section 3.2**)
4. Benchmarking models' implementations. (supplements **Section 5**)
5. Analysis of model performance based on region size vs. IoU. (supplements **Section 5.1**)
6. Analysis of granularity uni/n-gram frequency in Llama training data. (supplements **section 5.1**)
7. Adversarial prompting experiments for ViP-Llava 13B. (supplements **Section 5.1**)
8. Qualitative results from benchmarked models. (supplements **Section 5.1)**

2 SPIN Dataset Creation

2.1 Candidate Subpart Taxonomy

To identify candidate subpart categories for each object-part pair in PartImageNet [10], we prompted GPT-4 [1] with "Please list the canonical subparts of a <object>-<part>. Only include subparts that are clearly visible and recognizable to a layperson." The results were the following:

- **Quadruped-Head:** ears, eyes, nose, mouth, tongue, teeth, whiskers, forehead, cheeks, chin
- **Quadruped-Torso:** shoulders, back, belly, chest, ribs
- **Quadruped-Foot (leg):** hip, thigh, knee, shin, ankle, foot, toes, claws, pads, hoof
- **Quadruped-Tail:** base, midsection, tip
- **Biped-Head:** ears, eyes, nose, mouth, tongue, teeth, cheeks, forehead, chin, hair
- **Biped-Torso:** shoulders, chest, back, abdomen, waist, hips

- **Biped-Arm (includes hand):** shoulder, upper arm, elbow, forearm, wrist, hand, fingers, thumb
- **Biped-Foot (includes leg):** hip, thigh, knee, calf, ankle, foot, toes
- **Biped-Tail:** base, midsection, tip
- **Fish-Head:** eyes, mouth, gills, nostrils
- **Fish-Torso:** scales, lateral line, dorsal surface, ventral surface
- **Fish-Fin:** rays, spines, lobes, base
- **Fish-Tail:** caudal peduncle, caudal fin, upper lobe, lower lobe
- **Bird-Head:** beak, eyes, nostrils, ears, crown, nape
- **Bird-Torso:** chest, belly, back, flanks
- **Bird-Wing:** primaries, secondaries, coverts, alula
- **Bird-Foot (includes leg):** thighs, knees, shanks, toes, talons
- **Bird-Tail:** rectrices, pygostyle
- **Snake-Head:** eyes, mouth, nostrils, fangs, tongue
- **Snake-Torso:** scales, ventral plates, dorsal surface
- **Reptile-Head:** eyes, mouth, nostrils, tongue, teeth, ears
- **Reptile-Torso:** scales, belly, back, sides
- **Reptile-Foot (includes leg):** thigh, knee, ankle, toes, claws
- **Reptile-Tail:** base, midsection, tip
- **Car-Body:** hood, trunk, roof, doors, windows, fenders, bumpers
- **Car-Tire (includes all of the car wheel):** tread, sidewall, bead, rim, hubcap, valve stem
- **Car-Side-Mirror:** mirror glass, housing, adjustment mechanism
- **Bicycle-Head:** handlebars, stem, fork, front brake
- **Bicycle-Body:** frame, chain, pedals, crankset, gears
- **Bicycle-Seat:** saddle, seat post, clamp
- **Bicycle-Tire (includes all of the wheel):** tread, sidewall, tube, rim, spokes, hub
- **Boat-Body:** hull, deck, keel, rudder, bow, stern
- **Boat-Sail:** mainsail, jib, boom, mast, rigging
- **Aeroplane-Head:** cockpit, nose, windshield, radome
- **Aeroplane-Body:** fuselage, cabin, cargo hold, doors, windows
- **Aeroplane-Wing:** flaps, ailerons, slats, wingtips
- **Aeroplane-Tail:** vertical stabilizer, horizontal stabilizer, rudder, elevators
- **Aeroplane-Engine:** turbine, fan blades, exhaust, nacelle
- **Bottle-Body:** main chamber, label, base
- **Bottle-Mouth:** opening, neck, lip, cap

2.2 Final Subpart Taxonomy

As described in the main paper, we manually edited the results from GPT-4 to finalize the taxonomy. The final resulting taxonomy is as follows (parent objects listed in bold, followed by each part and its associated subparts):

- **Aeroplane** → Head: nosecone and windshield. Body: windows, doors, windshield, and decals. Wing: body and flaps. Engine: intake, outer casing, propeller, and cap. Tail: rudder, vertical stabilizer, horizontal stabilizer, and decals.

- **Bottle** → Body: label, shoulder, base, and neck. Bottle-mouth: rim and cap.
- **Boat** → Body: cockpit, deck, hull, bowsprit, decals, pontoon, and window. Sail: vertical beam, horizontal beam, decals, and sail.
- **Bicycle** → Head: handlebars, brake levers, headlight, bell or horn, grips, mirror, and tassel. Body: seat tube, top tube, down tube, head tube, fork, chainring, pedals, cranks, suspension, foot rest, stem, fender, axle, light, and parental control handle. Tire: tire, rim, spokes, fork and hub.
- **Biped** → Head: eyes, ears, nose, mouth, teeth, forehead, jaw, and neck. Torso: chest, abdomen, back, and shoulders. Arm: forearm, elbow, upper arm, wrist, palm, dorsal area, fingers, and shoulders. Foot: toes, heel, sole, and dorsal area.
- **Bird** → Head: eyes, beak, nostrils, forehead, neck, and cheek. Torso: breast, back, and belly. Foot: toes, claws, shank/forearm, thigh, knee, webbing, and ankle.
- **Car** → Body: door, window, roof, hood, trunk, bumper, decal, light, siren, grille, fender, windshield, windshield wiper, license plate, spoiler, exhaust, roll cage, ladder, plow, seat, hopper, trailer, and spare wheel. Tire: rim, tire, and hub cap. Side-mirror: mirror glass, housing, and mount.
- **Fish** → Head: eyes, mouth, gills, snout, and neck. Torso: neck, dorsal surface, ventral surface, and side. Fin: dorsal fins, pectoral fins, and ventral fins. Tail: lower lobe and upper lobe
- **Quadruped** → Head: eyes, ears, nose, mouth, horns, tusk, forehead, cheek, neck, and snout. Torso: back, chest, belly, side, shoulders, and neck. Foot: toes/hoof, claws, pads, dorsal area, heel, shank/forearm, knee/elbow, thigh/upper arm, and wrist/ankle.
- **Reptile** → Head: eyes, mouth, nostrils, tongue, neck, forehead, ears, casque, hood, and throat pouch. Torso: shell, belly, side, back, neck and dorsal fin. Foot: toes, webbing, pads, shank/forearm, knee, thigh/upper arm, wrist/ankle, and fin.
- **Snake** → Head: eyes, mouth, horn, nostrils, tongue, hood, forehead, and cheek. Torso: belly, back, and rattler.

2.3 PartImageNet Filtering

We removed the 29 images from PartImageNet with only the background class annotated (*i.e.*, no part annotations) because they couldn't support subpart annotation. We also excluded the following six PartImageNet's part classes that have ambiguous subpart decompositions: biped tails, bird tails, quadruped tails, bird wings, and bicycle seats.

Next, we restricted every PartImageNet category to include at most 1,200 images by using stratified sampling to preserve PartImageNet's original train, validation, and test split distribution. When sampling, we prioritized images containing the most parts from the part taxonomy to enhance the amount and diversity of annotated subpart annotations.

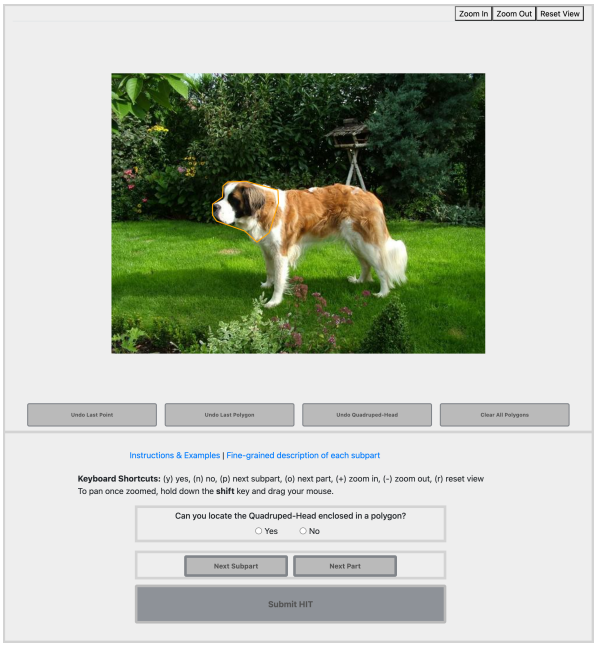

Fig. 1: Interface AMT crowdworkers used to create SPIN's ground truth annotations.

3 Crowdsourcing

3.1 Annotation Tool

Fig. 1 provides a screenshot of our crowdsourcing interface. We included in its design zooming functionality to enable more precise boundary annotations for subparts occupying tiny portions of images.

3.2 Crowdsourcing Implementation

We encouraged high-quality results in multiple ways. First, every annotator had to complete an initial onboarding task by passing a qualification test with five challenging annotation scenarios. Afterward, we provided a link to a 25-page PPT presentation that provided both generic annotation instructions (matching closely what they already used for their previous object-part annotation task with our team) as well as task-specific instructions clarifying for each super-category, textually and visually, how to annotate each subpart. These can be found at `https://joshmyersdean.github.io/spin/index.html`. After releasing tasks to AMT, we kept a live dialogue channel open with all annotators both by answering questions through email as well as via regular open Zoom sessions that individuals could join to solicit input. To further control quality, we released tasks to AMT in a phased rollout where we released all tasks for

a single super-category (*e.g.*, Quadruped, Bicycle) in a series of small batches before moving to the next super-category so workers could sharpen and retain skills on each category before moving to the next one. Following the completion of initial mini-batches per super-category, we manually spot-checked the results for potential worker confusion and provided individual feedback as needed until we found no further concerns. Additionally, throughout the annotation process, we manually inspected suspicious results, such as when workers flagged many parts and subparts as not being present, had missing subpart segmentations, or were outliers in the amount of time they took to complete tasks. We replaced unsuitable annotations as needed in addition to two authors inspecting every annotation and performing corrections as needed.

Toward's providing equitable compensation, we based HIT reward amounts on the maximum number of subparts a worker could encounter when annotating a particular super category. This design choice addressed the issue that there is high variation in the number of possible subparts per object category. To determine the pay amount, we conducted in-house testing to find the mean task duration relative to each super category. We found that paying 10 cents per subpart resulted in compensation above the United State's federal minimum wage. This rate resulted in compensating workers $1.10 per image for less complex categories like Boat, which only featured eleven potential subparts, versus $2.80 or $2.90 per image for more complex categories like Bicycles and Cars, respectively.

4 SPIN Analysis

4.1 Prevalence of Subpart Annotations per Part Category

We next characterize the subparts we augmented to the dataset by computing the frequency of subpart annotations per part category across SPIN's 11 supercategories with results shown in Fig. 2.

We also characterize the subparts we augmented to the dataset by computing the frequency of subpart annotations per part category across SPIN's 11 supercategories, with results shown in Fig. 2. We observe that car bodies exhibit the most subpart annotations per part category. We attribute this finding to the fact that the car body part category features the highest concentration of subpart categories (23 subparts) relative to all other part categories in SPIN. Additionally, the subpart categories within the car body part category, such as door, window, bumper, decal, and lights, often require multiple annotations per subpart. We observe similar trends in quadruped, biped, and reptile heads. Although this part category features fewer subparts than car bodies, they each contain subpart categories that often require multiple annotations to entirely segment, such as eyes, ears, nostrils, and cheeks. We also find that many of SPIN's images relative to these particular supercategories are biased toward these specific parts as they are often the principal area of focus in the image. For example, a reptile's feet could feature 20 toes and claws, yet a reptile's feet are unlikely to be the focus of an image. Last, these part categories also belong to super categories featuring

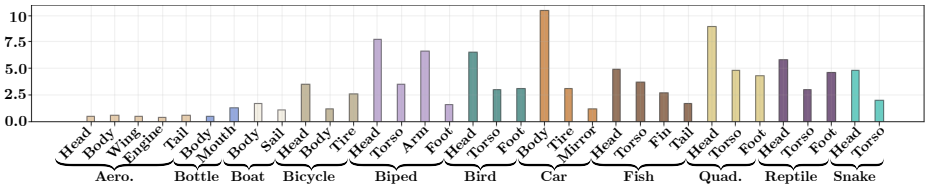

Fig. 2: Histogram visualizing the number of subpart-part category occurrences (in the thousands) across the SPIN dataset spanning each of the 34 part categories. We note that the biped and quadruped head, and the car body feature the most significant number of subpart occurrences within their parent part. (Aero=Aeroplane; Quad=Quadruped)

1200 images. In contrast, supercategories like aeroplane, bottle, and boat feature 311, 483, and 559 images, naturally lending them fewer subpart annotations than bipeds, quadrupeds, and cars.

4.2 Presence of Holes in Subparts

We evaluate the presence of holes within individual subparts in SPIN. For each subpart, we count how many holes it contains, defined by a polygon embedded within another.

Overall, we observe a relatively low presence of holes within subparts, with only 2.86% of subparts containing holes. Cars have the largest proportion of subparts containing holes at 13.54% and bottles have the lowest number of holes at 0.11%. Intuitively, a car contains subparts that naturally have holes, such as tires (which rims and hubcaps reside within), as well as grilles (which license plates and headlights reside within). In total, 6/11 object categories contain subparts in which greater than 1% contain holes: Aeroplane (3.58%), Bicycle (2.76%), Boat (6.85%), Car (13.54%), Reptile (1.19%), and Snake (4.70%). Of the remaining 5 object categories, all contain subparts with less than 1% having holes: Biped (0.56%), Bird (0.64%), Bottle (0.11%), Fish (0.12%), and Quadruped (0.54%).

Among *all subpart instances containing holes*, all have an average of less than 2. Boat has the highest average at 1.79 holes, and Fish has the lowest at 1.00 holes. A contributing reason for a scarcity of holes within subparts is that subparts are the finest level of granularity within an object, and thus other subparts typically do not reside within a subpart to create a hole.

4.3 Multiple Polygons in Subparts

The prevalence of requiring multiple polygons per subpart is shown with respect to objects in Fig. 3.

We find that 31.21% (33,188) of subpart annotations have more than one polygon. In other words, subpart categories belonging to these object categories contain multiple polygons in the semantic annotations. Most subpart occurrences requiring multiple polygons occur for biped arms, quadruped and reptile feet,

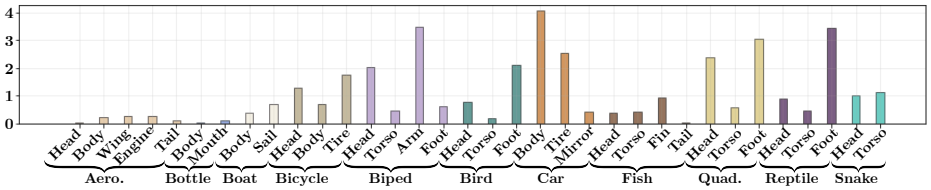

Fig. 3: Histogram visualizing the number of subparts (in the thousands) that required multiple polygons to annotate spanning each of the 34 part categories. We note that biped arms, quadruped and reptile feet, and car bodies feature the most subpart occurrences requiring multiple polygons to annotate. (Aero=Aeroplane; Quad=Quadruped)

and car bodies. We attribute this finding to the intrinsic properties of these particular subparts and the viewing angle. For instance, the biped arms, reptile, and quadruped feet often exhibit 5-10 fingers and toes, and reptile and quadruped feet sometimes feature claws that can require an additional 5-10 polygons. In addition, car bodies can contain 2-20 windows depending on the vehicle type, as well as 2 lights and 4 tires, underscoring why this category has the most significant number of multi-polygon subpart annotations of all object categories.

We next characterize subparts consisting of multiple polygons based on two metrics: 1) *Extent*: the ratio of a subpart's area to it's bounding box. Values are in (0,1], where values approaching 0 mean that a contour occupies little area in it's bounding box (*e.g.*, a thin diagonal line) and 1 means that a contour is perfectly contained (*e.g.*, a square).; and *Boundary complexity*: ratio of a subpart's area to the length of its perimeter (*i.e.*, isoperimetric quotient). Values range from 0 (highly jagged boundary) to 1 (circular). For regions consisting of multiple polygons, we record the mean of each metric for each polygon. We compute the average of each metric across all constituent polygons in a subpart's annotation. Results are shown in Figure 4.

Regarding shape in single—and multi-polygon subpart annotations, the primary trend we observe is that single-polygon annotations take up the majority of their bounding boxy. In contrast, multi-polygon annotations tend to only occupy 50% of their bounding box (*i.e.*, Fig. 4 a, b, values closer to 1 compared to b). Intuitively, single polygons may take up more space as there is less background captured in the bounding box (*i.e.*, there are less overall background pixels).

We also see a similar trend in boundary complexity, especially in Bicycles, as their respective inter-quartile ranges get much wider and further away from 0.5 in multi-polygon part annotations compared to single-polygon subpart annotations, ultimately exhibiting moderate albeit more complex boundary complexity among multi-polygon subpart annotation versus single-polygon subpart annotations (*i.e.*, Fig. 4 c, d). We see this trend in bicycles more than in cars because subparts like tires on a bicycle occupy a more significant portion of the bicycle's area compared to a tire on a vehicle, which occupies much less area relative to the object.

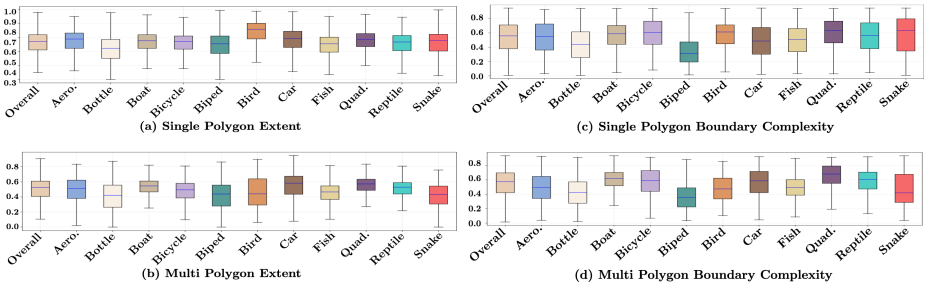

Fig. 4: Subpart extent and boundary complexity relative to the number of polygons required to segment the subpart, grouped by their respective super categories.

5 Model Benchmarking

5.1 Design of Benchmarked Models

For each benchmarked model, we report the number of parameters, visual encoder, LLM (*i.e.*, text encoder), capabilities, and model source for inference in Table 1. For SAM [12], we adopt the commonly used ViT-H [8] variant. For models producing bounding boxes, we post-process predicted object detections for each semantic category by converting them into a single pixel-wise mask to create a semantic segmentation.

We also report the specific prompts used for each model, as they vary with each model's official implementation. For objects, <region> is the name of the object (*e.g.*, quadruped, antelope) and for (sub)parts, <region> is the name of the (sub)part and the object (*e.g.*, eyes of the quadruped, eyes of the antelope). We use the same prompts for models that have both 7B and 13B variants (Ferret, LISA, PixelLLM, ViP-Llava).

Open-Vocabulary Localization Prompts.

- **Ferret:** "Please locate the <region> in this image. Only locate the <part> but locate all instances of the <part>." We omit the second sentence when doing object-level localization.
- **CoGVLM:** "Please describe the <object> in detail and provide its coordinates [[x0, y0, x1, y1]]."
- **Shikra:** "Can you point out <region> in the image <image> and provide the coordinates of its location?" Where <image> is the tokenized image.
- **Kosmos2:** "<grounding><phrase> the <region> </phrase>"
- **LISA:** "Please segment the <region> in this image."
- **GLaMM:** "Please segment the <region> in this image."
- **PixelLLM:** "Please segment the <region> in this image."

Interactive Understanding Prompts.

- **Kosmos2:** "<phrase>Is there a <region> in the image? Think step-by-step."

Model	Parameters	Visual Encoder	LLM	Open-Vocab Localization	Interactive Understanding	Model Source
HIPIE [24]	200M	ResNet-50 [11]	BERT [7]	✓	✗	github.com/berkeley-hipie/HIPIE
HIPIE [24]	800M	ViT-H [8]	BERT [7]	✓	✗	github.com/berkeley-hipie/HIPIE
LISA [13]	7B	SAM ViT-H [12]	LLaVa-7B-v1-1 [14]	✓	✗	github.com/dvlab-research/LISA
LISA [13]	13B	SAM ViT-H [12]	Llama-2-7B [21]	✓	✗	github.com/dvlab-research/LISA
GLaMM [18]	7B	SAM ViT-H [12]	Vicuna-7B [5]	✓	✗	github.com/mbzuai-oryx/groundingLMM
PixelLLM [19]	7B	CLIP-ViT-L/14 [17]	LlaVA-7B [14]	✓	✗	github.com/MaverickRen/PixelLM
PixelLLM [19]	13B	CLIP-ViT-L/14 [17]	LlaVA-llama-13B [14, 21]	✓	✗	github.com/MaverickRen/PixelLM
CoGVLM [23]	17B	EVA2-CLIP-E [9]	Vicuna1.5-7B [5]	✓	✓	github.com/THUDM/CogVLM
Ferret [25]	7B	CLIP-ViT-L/14 [17]	Vicuna-7B [5]	✓	✓	github.com/apple/ml-ferret
Ferret [25]	13B	CLIP-ViT-L/14 [17]	Vicuna-13B [5]	✓	✓	github.com/apple/ml-ferret
Shikra [4]	7B	CLIP-ViT-L/14 [17]	Vicuna-7B [5]	✓	✓	github.com/shikras/shikra
Kosmos2 [16]	1.6B	Unspecified	MAGNETO Transformer [22]	✓	✓	huggingface.co/docs/transformers/en/model_doc/kosmos-2
ViP-Llava [3]	7B	CLIP-ViT-L/14 [17]	Vicuna-7B [5]	✓	✓	huggingface.co/docs/transformers/main/en/model_doc/vipllava
ViP-Llava [3]	13B	CLIP-ViT-L/14 [17]	Vicuna-13B [5]	✓	✓	huggingface.co/docs/transformers/main/en/model_doc/vipllava
Osprey [26]	7B	CLIP-ConvNeXt-L [17]	Vicuna-7B [5]	✗	✓	github.com/CircleRadon/Osprey

Table 1: Overview of benchmarked foundation models with respect to their parameters, encoder types, LLM, task capabilities, and model source. (B=billions; M=millions).

- **Ferret:** "Is this <mask><pos> a <region>? Only answer yes or no with no other output." Where <mask><pos> is the tokenized mask with positional encoding.
- **Osprey:** "Is this <mask><pos> a <region>? Only answer yes or no with no other output." Where <mask><pos> is the tokenized mask with positional encoding.
- **Ferret:** "Is this <mask><pos> a <region>? Only answer yes or no with no other output." Where <mask><pos> is the tokenized mask with positional encoding.
- **ViP-Llava:** "Is there a <region> in the blue region? Answer yes or no." Where "blue region" is the overlayed ground truth segmentation mask of the region.
- **Shikra:** "For this image <image>, I want a simple and direct yes or no answer to my question: Is there a <region> in this region <boxes>?" in which <image> is the tokenized image, and <boxes> is the ground truth bounding box.

5.2 HIPIE Analysis

Despite poor localization from HIPIE, it is worth noting that HIPIE has interesting hierarchical performance results. First, it achieved nearly perfect spatial consistency between parts and objects (*i.e.*, SpCS-P2O) and perfect spatial consistency between subparts and parts (*i.e.*, SpCS-S2P). In other words, when HIPIE predicted parts are always perfectly contained within their parent parts which, in turn, are typically perfectly contained within their parent objects. When examining HIPIE's semantic consistency with SeCS metrics for *general* and *specific* categories, we find ResNet-50 outperforms ViT-H for general categories (85.85% vs. 73.38% SeCS) despite ViT-H's higher object mIoU. This suggests that ViT-H's increased computational power does not enhance part/subpart accuracy, but rather only object-level performance. ViT-H also shows a higher abstention rate from subpart predictions (*i.e.*, does not predict segmentations) than ResNet-50 (35.77% vs. 24.71%). For specific categories, both backbones score high on SeCS (94.58% for ResNet-50 and 100% for ViT-H) but abstain 84% of the time, likely

due to the large, similar *specific* category list (154 specific vs. 11 general categories), highlighting issues like differentiating 'box turtle' from 'mud turtle' in specific categories. In contrast, all labels in the general categories share little similarity.

	Object		Part		Subpart	
Model	R^2	**p-value**	R^2	**p-value**	R^2	**p-value**
HIPIE R50	0.00		0.00		0.00	
HIPIE ViT-H	0.00		0.00		0.00	
PixelLLM 7B [19]	0.00		0.67		0.44	
PixelLLM 13B [19]	0.00		0.67		0.37	
LISA 7B [13]	0.04		0.63		0.28	
LISA 13B [13]	0.01		0.62		0.49	
GLaMM [18]	0.01		0.55		0.35	
Ferret 7B [25]	0.11		0.64		0.28	
Ferret 13B [25]	0.04		0.65		0.35	
CoGVLM [23]	0.05		0.01		0.10	
Shikra [4]	0.07		0.04		0.20	
Kosmos2 [16]	0.07		0.76		0.36	

Table 2: Impact of size on predicting IoU for open-vocabulary localization models. We report Pearson R^2 coefficients and p-values. Blue cells represent statistically significant results for $\hat{\beta}_1$ in IoU $\sim \hat{\beta}_1 \log(\text{region size}) + \hat{\beta}_0$ ($p < 0.001$), and orange represents results that are not statistically significant. Above the dashed line represents segmentation models, and below represents models that output bounding boxes.

6 Analysis of Region Size vs. IoU

To examine the influence that region size has on segmentation results, we ran a linear regression, IoU $\sim \hat{\beta}_1 \log(\text{region size}) + \hat{\beta}_0$, and calculated the Pearson R^2 correlation coefficients for each model at every granularity level, also noting the median p-value of $\hat{\beta}_1$ to assess the significance of region size on IoU performance. Results are shown in Table 2. We include HIPIE in the table but exclude it in our discussion as its poor results skew trends.

Overall, we observe mixed outcomes. No significant positive correlation is observed for objects, with Pearson correlation R^2 values between 0.003 and 0.105 (median p-value ≈ 0.009), suggesting that an object's segmentation size does not strongly predict IoU scores. Conversely, a positive correlation is noted for parts, indicated by R^2 values ranging from 0.014 to 0.759 (median p-value $\approx$ 1e$-$10), implying that larger parts may correspond to higher IoU scores, depending on the model. Subparts show a weaker positive correlation, with R^2 values from 0.102 to 0.491 (median p-value $\approx$ 2e$-$20), highlighting that while segmentation size impacts performance, it is not the predominant factor.

7 Analysis of granularity uni/n-gram frequency in Llama training data

Given the proprietary nature of Llama's training data, we utilize RedPajama [6], a 1.4 trillion token corpus designed to closely replicate Llama's dataset, as a stand-in. We use the Llama tokenizer for tokenization and examine occurrences of uni-grams across three categories: subparts ($N = 206$), parts ($N = 40$), and objects ($N = 11$). We leverage the ∞-gram [15] API for counting these occurrences within the RedPajama dataset. We observe decline in average uni-gram occurrence frequency when increasing granularity (*e.g.*, part to subpart). This trend is depicted in Fig. 5a. Further analysis of parts and subparts n-grams (Fig. 5b) reveals that subpart n-grams (*e.g.*, 'eyes of the quadruped') are significantly less frequent, with an average of 7 instances, compared to parts (*e.g.*, 'head of the quadruped'), which average 75 instances.

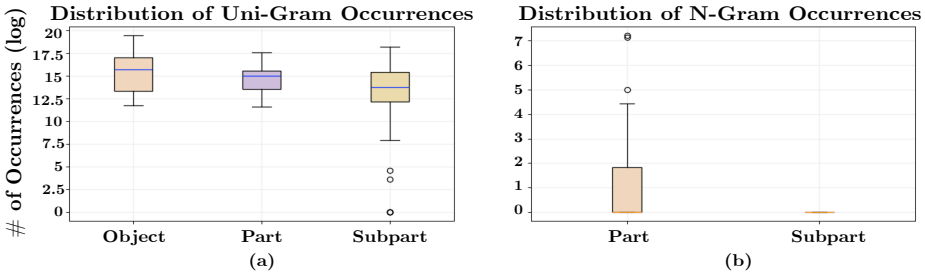

Fig. 5: (a) Distribution of uni-gram (*e.g.*, quadruped, head, eyes) across the RedPajama dataset for objects, parts, and subparts. (b) Distribution of n-gram (*e.g.*, head of the quadruped, eyes of the quadruped) across the RedPajama dataset for parts and subparts. We show a log scale to account for wide-range values.

8 ViP-Llava Adversarial Prompting

We conducted two different adversarial prompting experiments for ViP-Llava 13B to better understand its near-perfect performance on interactive understanding. First, we conducted an adversarial experiment where we prompted the model the same way as the original experiment but randomly swapped out the object category for a different one among our set of object super-categories. As a consequence, the answer to the question, `"Is there an <object> in this <region>?"` is always 'no.' We observe for this experiment that within object categories, mean accuracy decreases to 73.26% (-25.07pp) for objects, 98.08% (-1.82pp) for parts, and 96.62% (-2.73pp) for subparts. These findings suggest that the inclusion of granular phrases (*e.g.*, cheek) can help calibrate a model's confidence and reduce hallucinations, potentially due to the intrinsic associations it may make (*e.g.*, recognizing that a bicycle does not have a cheek).

Second, we prompted the model with the negation of the original prompt, "Is there not an <object> in this <region>?", in which the answer is always 'no'. Overall, we observe large decrease in performance with a mean accuracy of 28.36% (13.65% specific) for objects, 3.23% (7.33% specific) for parts, and 4.74% (4.17% specific) for subparts. This big difference in performance from the results in the main paper reinforces findings from prior work that models struggle with negation [2]. Moreover, these results highlight the importance of adversarial prompting and red-teaming foundation models to probe their biases (*e.g.*, through tools like VLSlice [20]), such as a predisposition to answering yes to content that is not present within an image.

9 Qualitative Results

9.1 Foundation Model Results

Qualitative results for open-vocabulary object localization models are shown for 5 diverse examples in Fig. 6 (segmentation) and Fig. 7 (object detection). We show examples for tiny subparts (*i.e.*, eyes of the snake, nostrils of the bird) and large subparts (*i.e.*, horns of the quadruped, neck of the bottle, grille of the car).

For models capable of segmentation (LISA 7/13B, PixelLLM 7/13B, GLaMM), varied results are observed across examples. LISA 7B localizes snake eyes most accurately, while others locate the entire head or body portions. No model precisely segments bird nostrils, with the closest attempts segmenting the beak. Only LISA variants perfectly segment antelope horns without additional regions. For the grille, GLaMM provides the best segmentation, albeit with missing cruft. Regarding the bottleneck, LISA 13B achieves near-precise segmentation (aside from the inclusion of the shoulder), whereas other models either segment partial regions (LISA 7B, PixelLLM 7/13B) or all regions except the main label on the bottle (GLaMM).

For models that produce bounding boxes (CoGVLM, Ferret 7/13B, Shikra, Kosmos2), relatively consistent results are observed across examples. CoGVLM precisely locates tiny subparts (eye, nostril), while others produce shifted or object-encompassing bounding boxes. For larger regions (grille, horns, neck), all models except Shikra correctly localize antelope horns, and all except Kosmos2 accurately locate the car grille. Conversely, Shikra provides the closest bounding box for the bottleneck, with other models only capturing partial or complete bottle regions.

Overall, these results support our quantitative findings, with all models generally performing poorly on subpart localization. Overall, CoGVLM produces the best results, aligning with its superior quantitative performance.

9.2 HIPIE Results

Qualitative results for predicted subparts by HIPIE are shown in Fig. 8. (a) Shows a partially correct segmentation of bicycle handlebars, with incorrect

labeling of the rest as "fender". (b) Demonstrates an out-of-distribution sample with incorrect labeling. (c) Exhibits a small number of correct class labels ("door" and "tire") but with inaccurate segmentations. (d) Displays a semantically incoherent combination of "bird back" and "fish eyes", highlighting the need for holistic evaluation of granular segmentations (*e.g.*, our proposed consistency scores). Overall, these poor results corroborate the quantitative findings reported in the main paper.

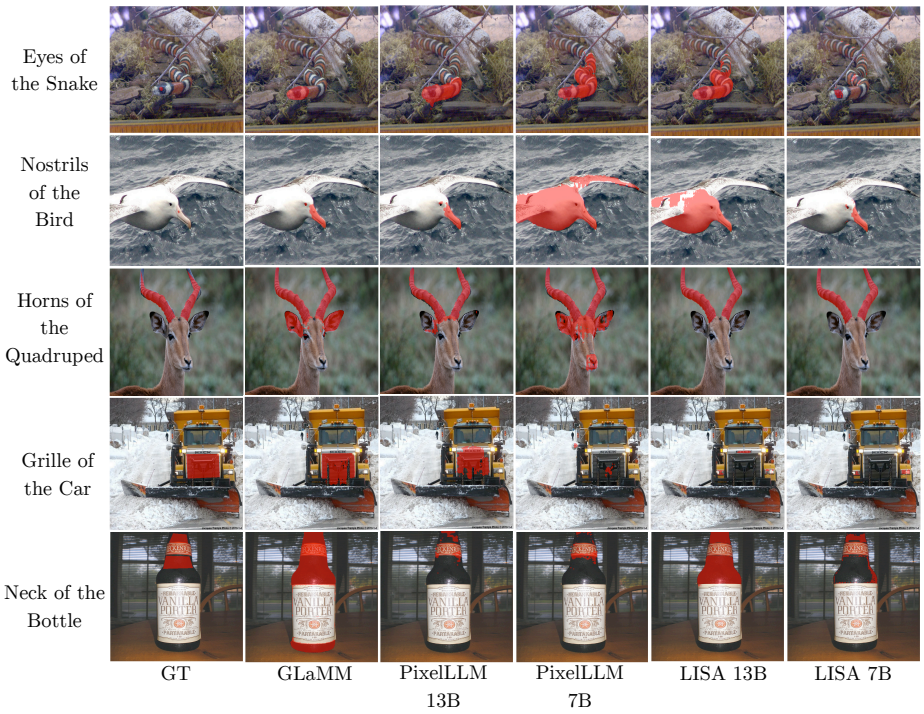

Fig. 6: Qualitative results of models producing segmentation predictions, shown in red. Each row represents a different subpart. Columns display, from left to right: ground truth segmentations, followed by predictions from each method. For visualization purposes, all images are resized to square aspect ratios.

Fig. 7: Qualitative results of models producing bounding box predictions, shown in red. Each row represents a different subpart. Columns display, from left to right: ground truth bounding boxes, followed by predictions from each method. For visualization purposes, all images are resized to square aspect ratios.

Fig. 8: Qualitative results from HIPIE, with each panel showing all predicted segmentations with their corresponding label classification depicted in the same color. For visualization purposes, all images are resized to square aspect ratios.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Berglund, L., Tong, M., Kaufmann, M., Balesni, M., Stickland, A.C., Korbak, T., Evans, O.: The reversal curse: Llms trained on" a is b" fail to learn" b is a". arXiv preprint arXiv:2309.12288 (2023)
3. Cai, M., Liu, H., Mustikovela, S.K., Meyer, G.P., Chai, Y., Park, D., Lee, Y.J.: Making large multimodal models understand arbitrary visual prompts. In: IEEE Conference on Computer Vision and Pattern Recognition (2024)
4. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: Unleashing multimodal llm's referential dialogue magic. arXiv preprint arXiv:2306.15195 (2023)
5. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (March 2023), `https://lmsys.org/blog/2023-03-30-vicuna/`
6. Computer, T.: Redpajama: an open dataset for training large language models (October 2023), `https://github.com/togethercomputer/RedPajama-Data`
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
9. Fang, Y., Sun, Q., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva-02: A visual representation for neon genesis. arXiv preprint arXiv:2303.11331 (2023)
10. He, J., Yang, S., Yang, S., Kortylewski, A., Yuan, X., Chen, J.N., Liu, S., Yang, C., Yu, Q., Yuille, A.: Partimagenet: A large, high-quality dataset of parts. In: European Conference on Computer Vision. pp. 128–145. Springer (2022)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 4015–4026 (October 2023)
13. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. arXiv preprint arXiv:2308.00692 (2023)
14. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023)
15. Liu, J., Min, S., Zettlemoyer, L., Choi, Y., Hajishirzi, H.: Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. arXiv preprint arXiv:2401.17377 (2024)
16. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Wei, F.: Kosmos-2: Grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824 (2023)
17. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from

natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
18. Rasheed, H., Maaz, M., Shaji, S., Shaker, A., Khan, S., Cholakkal, H., Anwer, R.M., Xing, E., Yang, M.H., Khan, F.S.: Glamm: Pixel grounding large multimodal model. arXiv preprint arXiv:2311.03356 (2023)
19. Ren, Z., Huang, Z., Wei, Y., Zhao, Y., Fu, D., Feng, J., Jin, X.: Pixellm: Pixel reasoning with large multimodal model (2023)
20. Slyman, E., Kahng, M., Lee, S.: Vlslice: Interactive vision-and-language slice discovery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15291–15301 (2023)
21. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models (2023)
22. Wang, H., Ma, S., Huang, S., Dong, L., Wang, W., Peng, Z., Wu, Y., Bajaj, P., Singhal, S., Benhaim, A., et al.: Magneto: a foundation transformer. In: International Conference on Machine Learning. pp. 36077–36092. PMLR (2023)
23. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., et al.: Cogvlm: Visual expert for pretrained language models. arXiv preprint arXiv:2311.03079 (2023)
24. Wang, X., Li, S., Kallidromitis, K., Kato, Y., Kozuka, K., Darrell, T.: Hierarchical open-vocabulary universal image segmentation. Advances in Neural Information Processing Systems **36** (2024)
25. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.F., Yang, Y.: Ferret: Refer and ground anything anywhere at any granularity. arXiv preprint arXiv:2310.07704 (2023)
26. Yuan, Y., Li, W., Liu, J., Tang, D., Luo, X., Qin, C., Zhang, L., Zhu, J.: Osprey: Pixel understanding with visual instruction tuning (2023)