CALICO: Part-Focused Semantic Co-Segmentation with Large Vision-Language Models

Supplementary Material

A. MIXEDPARTS

A.1. Single-Image Part Segmentation Datasets

To construct a robust dataset for part-focused semantic co-segmentation, we carefully curate generalizable and diverse data at various levels of detail. This entails selecting datasets that cover a wide range of objects and parts, both rigid (*e.g.*, utensils, vehicles, scissors) and non-rigid (*e.g.*, animals, humans), while not being too domain-specific (*e.g.*, birds or celebrities' faces), which could potentially encourage overfitting or introduce bias in training. Therefore, we select the following datasets to construct MIXEDPARTS:

- **PartImageNet** [18] is a high-quality part-focused extension of ImageNet [13] covering a variety of object classes with mostly animals (*e.g.* bird, fish, *etc.*) to facilitate non-rigid part understanding. Each image in PartImageNet contains only one foreground object, which can encourage the model to focus on important foreground objects when comparing two images. Prior to use, we make modifications to the original dataset to enhance generalizability, as detailed in Appendix A.2.
- **ADE20K-Part234** [65] is a revised version of the ADE20K scene parsing dataset [77, 78] with an emphasis on object parts. The original ADE20K dataset encompasses a wide variety of scenes, including indoor, outdoor, and urban environments, which naturally lends itself to more complex visual signals covering multiple objects in contrast to PartImageNet. However, less than 15% of the dataset contains part annotations; in addition, some part labels are too granular, which may encourage overfitting while not being too beneficial for general part understanding (*e.g.* "table stretcher" and "table h-stretcher"). To amend these disadvantages, Wei et al. [65] introduces ADE20K-Part234, a clean, part-focused version of ADE20K for improved part analysis.
- **PACO-LVIS** [49] is a part-centric version of LVIS [17] that is based on COCO [34] and focuses on diverse everyday objects, further contributing to the diversity of object categories in MIXEDPARTS. PACO contains an extensive list of object-part categories as well as complex images with multiple objects and parts, providing finer granularity for part understanding.

To curate MIXEDPARTS, we first select 1,885 pairs of object categories across all 3 datasets that have at least one common part label (*e.g.* "armchair's seat" and "swivel chair's seat") to ensure annotation availability. However, due to the ambiguity of natural language, the same part name can refer to object parts that are not typically intuitively comparable. For example, even though both a bus and a microwave oven may have a door, they are not commonly compared. Therefore, we manually curate intuitively comparable object pairs from all possible pairs, resulting in 964 pairs of categories across all 3 datasets.

With the object pairings available, we pair up individual images corresponding to our common object, common part, and unique part localization subtasks. For common object parts, we select images that have at least a common visible object and/or object part. However, since different object classes can share the same parts, we also include images of logically comparable objects of different classes, for instance, a chair and an ottoman (both seating furniture) or an airplane and a bird (both having wings and usually compared as flying objects). We ensure stratification of object categories in MIXEDPARTS to reflect the data distribution in the original datasets.

A.2. PartImageNet Modifications

While PartImageNet provides high-quality segmentation masks across a diverse range of object classes, its categories are often too abstract and not commonly used in natural language due to their broad scope. For instance, PartImageNet classifies all four-legged animals under the "quadruped" supercategory. Although technically accurate, such classifications are too generic for practical use in everyday language. To address this, we manually selected the most commonly used object class associated with each category provided by ImageNet. The dataset includes WordNet synset IDs (*e.g.*, "n02071294") that correspond to specific object classes.

Using these synset IDs, we extracted the hierarchical path of each class within the WordNet graph. For example, the synset ID "n02071294" maps to the following WordNet path: living_thing $\rightarrow$ organism $\rightarrow$ animal $\rightarrow$ chordate $\rightarrow$ vertebrate $\rightarrow$ mammal $\rightarrow$ placental $\rightarrow$ aquatic_mammal $\rightarrow$ cetacean $\rightarrow$ whale $\rightarrow$ toothed_whale $\rightarrow$ dolphin $\rightarrow$ killer_whale. From this path, we selected *whale* as the representative object category for this class. We repeat this process for all object classes provided by PartImageNet. The object classes we use alongside the original supercategory in parentheses are shown in Table 6.

B. MIXEDPARTS Dataset Statistics

Figure 9 and Table 3 provide a comprehensive overview of our benchmark, illustrating its structure and the distribution of objects and parts. The inner circle of the donut chart represents the entire dataset, comprising a total of 2,382,747 samples (image pairs), sourced from PartImageNet, PACO-LVIS, and ADE20K-Part234, each contributing to roughly a third of the total dataset. The outer ring of the chart further divides these sources into categories of objects and parts, contributing roughly half, ensuring balanced representation of various object–part relationships. In MIXEDPARTS, there are 2,509,552 instances of common objects, 3,443,758 instances of common parts, and 5,557,188 instances of unique parts. On average, each image pair includes 2 object instances (one per image), 4 common part instances, and 5.3 unique part instances. Sample complexity varies widely, with a maximum of 10 common objects, 56 common parts, and 72 unique parts per sample. Median counts are 2 for common objects, 4 for common parts, and 3 for unique parts, reflecting that a single object pair can involve multiple shared and unique parts. The dataset spans 141 object categories and 196 part categories, covering a wide variety of objects and parts, totaling 65,960 distinct object instances and 154,463 distinct part instances. Figures 11-12 present the top-30 most frequent object and part categories, while Figures 13-14 show the top-30 common and unique parts, respectively.

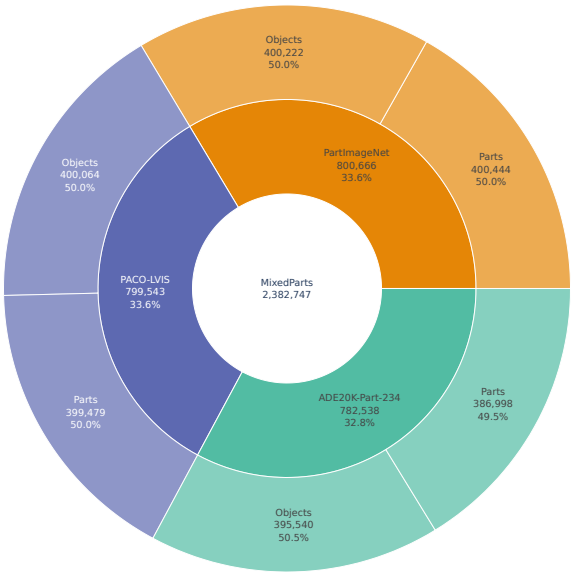

Figure 9. **MIXEDPARTS Dataset Overview.**

C. Additional Experimental Setup Details

C.1. Implementation Details

CALICO. We use PyTorch [46] to implement and optimize CALICO, with DeepSpeed [51] for efficient training. We initialize our model on GLaMM's [50] pretrained checkpoint,[1] the Q-Former module on InstructBLIP's [10] Vicuna-7B-aligned checkpoint,[2] and align them with 3.2M samples from GLaMM's GranD dataset. All training is conducted on four NVIDIA A40 GPUs with 48GB memory. LoRA layers are applied to the query and value projections, configured with a rank of 8 and a scaling factor of 16. We train for 10 epochs, each comprising 500 steps, using the AdamW optimizer [40] with a 4e-4 initial learning rate and beta coefficients set to 0.9 and 0.95, respectively. We warm up the learning rate over 100 training steps and then use a linear decay scheduler over the remaining training time. We employ a dropout rate of 0.05, 1.0 gradient clipping, and set the batch size to 4 alongside gradient accumulation every 4 steps. The loss coefficients are set to $\lambda_{\text{text}} = 1.0$, $\lambda_{\text{focal}} = 2.0$, and $\lambda_{\text{Dice}} = 0.5$. We retain all original implementation details for our baselines while keeping the effective batch size the same across all finetuned approaches to ensure fairness. We use DeepSpeed's profiler[3] to compute TFLOPS.

Baselines. We provide implementation details for our baselines, which include three zero-shot and two finetuned LVLM-based approaches:

- **Cascade:** To evaluate whether a simple modular pipeline can effectively solve part-focused semantic co-segmentation, we first design an cascaded approach that sequentially integrates strong zero-shot models, each excelling at different subcomponents of the task: (1) Sparkles [22], a powerful multi-image model, is used to identify relevant objects across images, followed by (2) GPT-4o API [1] to extract object-part relationships, and finally, (3) LISA [27], a segmentation-based LVLM, to produce segmentation masks for the identified objects and parts. Given an image pair, we first prompt Sparkles to list the objects or parts present across both images. Next, we prompt GPT-4o with this information to identify the common or unique objects or parts, depending on the task. Finally, we prompt LISA with the objects/parts extracted by GPT-4o to generate pixel-level segmentation masks. Prompt templates for all models of this cascaded approach are shown in Figure 10.
- **Multi-Image VLPart:** VLPart [60] is an open-vocabulary part segmentation model that can segment object parts at different granularities. It combines a conventional Mask R-CNN [19] with a modern Swin Transformer backbone [38] and a CLIP [48] text classifier for open-world image classification. Although VLPart cannot perform co-segmentation, its strong zero-shot

[1] https://huggingface.co/MBZUAI/GLaMM-GranD-Pretrained.

[2] https://storage.googleapis.com/sfr-vision-language-research/LAVIS/models/InstructBLIP/instruct_blip_vicuna7b_trimmed.pth.

[3] https://www.deepspeed.ai/tutorials/flops-profiler/.

Statistic	Common Objects	Common Parts	Unique Parts
Total # Instances	2,509,552	3,443,758	5,557,188
Average # Instances/Sample	2.099	4.037	5.269
Maximum # Instances/Sample	10	56	72
Median # Instances/Sample	2	4	3

Table 3. **MIXEDPARTS Statistics** on # instance of objects/parts. Here, an object/part instance in MIXEDPARTS refers to a single occurrence of an object or part in a sample, while a sample corresponds to an image pair.

object-part segmentation capabilities enables a simple baseline: performing segmentation on individual images and simply examining the common and unique predictions. Specifically, for a given image pair, we perform inference on both images and extract all predicted object and part masks. To identify common objects, we compare the sets of predicted objects across images. For common and unique parts, we aggregate the sets of all predicted parts and derive their associated objects. We follow the default configuration from the official repository[4], including the detection confidence threshold of 0.7 (*i.e.*, retaining predictions with confidence scores above 0.7) to initialize the model. We use the checkpoint with a Swin-base cascade backbone trained on the parsed ImageNet dataset.[5]

- **Multi-Image PartGLEE:** PartGLEE [32] is a recent foundation model that uses Q-Former to query object and part representations from large-scale training data, boasting better object-part segmentation capabilities than VLPart. For this baseline, we simply obtain single-image outputs from PartGLEE (100 masks per image), keeping only those with non-zero scores (55 on average). Similar to Multi-Image VLPart, we compare the predicted class labels across the image pair to identify common and unique objects and parts. We use PartGLEE's official implementation,[6], initializing with the released Swin-Large checkpoint and default settings.
- **Multi-Image LISA:** LISA [27] is an LVLM trained to perform referring segmentation, built on a Vicuna-7B backbone and SAM. Since LISA was not trained for multi-image processing, we replicate CALICO's multi-image implementation on the LISA codebase. We initialize LISA on their 7B checkpoint[7] and finetune the mask decoder alongside LoRA.
- **Multi-Image GLaMM:** GLaMM [50] is a strong single-image segmentation-based LVLM constructed for the Grounded Conversation Generation (GCG) task involving multi-turn pixel-grounded dialogue. GLaMM combines a Vicuna-based language backbone with SAM, similarly to LISA, and a novel RoIAlign-based region encoder. We implement multi-image processing for GLaMM by sequentially feeding the mask decoder with the encoded images and corresponding segmentation tokens to obtain segmentation masks. Both CALICO and Multi-Image GLaMM are initialized from GLaMM's pretrained model weights.[8] Since LISA and GLaMM are not natively designed to handle multiple images, we adapt CALICO's implementation for distinguishing segmentation tokens belonging to different images. This is achieved by appending image identifiers to the `[SEG]` tokens, *e.g.*, `(IMAGE1)`, enabling the models to process the image-specific token sets separately.

C.2. Evaluation Metrics

Our evaluation consists of two parts: (1) segmentation quality and (2) semantic label accuracy for objects and parts. To evaluate performance on segmentation tasks, we employ mean Intersection over Union (**mIoU**), a widely used metric that measures the average overlap between predicted and ground truth masks across all classes. We also employ Average Precision at a 50% IoU threshold (**AP50**), which measures the average precision across all recall levels at a fixed IoU threshold. Specifically, AP50 considers a predicted mask correct if its IoU with the ground truth mask is at least 50%, summarizing model performance at this specific threshold. Additionally, following GLaMM [50], we report **Recall**, which evaluates region-specific grounding by using a two-phase validation approach based on IoU and SentenceBERT [52] similarity thresholds. In particular, a prediction is considered a true positive if its mask has IoU $\geq 50\%$ with the ground truth and its text label exceeds 50% similarity with the ground truth.

To evaluate semantic labeling performance for the segmented objects and parts, we compute Semantic Similarity and Semantic IoU. Semantic Similarity (**SS**) measures similarity between predicted and true labels in the SentenceBERT [52] embedding space, following prior work [9, 71], while Semantic IoU (**S-IoU**) computes token-level overlap between predicted and ground truth labels, defined as

$$\text{S-IoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{|V(y_i) \cap V(\hat{y}_i)|}{|V(y_i) \cup V(\hat{y}_i)|},$$

where $V(y)$ denotes the set of words comprising label y.

[4] https://github.com/facebookresearch/VLPart.
[5] https://github.com/PeizeSun/VLPart/releases/download/v0.1/swinbase_cascade_lvis_paco_pascalpart_partimagenet_inparsed.pth.
[6] https://github.com/ProvenceStar/PartGLEE.git.
[7] https://huggingface.co/xinlai/LISA-7B-v1.
[8] https://huggingface.co/MBZUAI/GLaMM-GranD-Pretrained.

Method	Recall↑	# image tokens↓	TFLOPS↓	inference time↓			
				CO	CP	UP	All
Multi-Image GLaMM [50]	54.9	576	42.3	8.4	21.3	26.3	18.7
Multi-Image LISA [27]	55.5	256	44.0	5.5	14.9	18.9	13.1
CALICO (ours)	**59.7**	**32**	**28.5**	**4.3**	**10.2**	**12.3**	**9.1**

Table 4. **CALICO Efficiency in Numbers**. Alongside TFLOPS, we show inference time per sample in seconds for the common objects (CO), common parts (CP), unique parts (UP) tasks, as well as the average (All).

D. Additional Experiments

D.1. Computational Efficiency Evaluation

For multi-image tasks, computational efficiency is critical to ensure scalability in inference and training as the number of images increases. To address this, CALICO employs Q-Former to query image embeddings, reducing the number of tokens per image to just 32, which is 8 times fewer tokens than LISA (256 tokens) and 18 times fewer than GLaMM (576 tokens). As shown in Table 4, this reduces TFLOPs by 35.23% over LISA and 32.62% over GLaMM (LISA:44.0 *vs*. GLaMM:42.3 *vs*. CALICO:28.5 TFLOPs), yielding a $\sim$1.5$\times$speed-up and a 7.6% relative performance gain over LISA. These results demonstrate that CALICO offers substantial improvements in both computational efficiency and task performance compared to strong baselines.

D.2. Per-Task Experimental Results

Table 5 presents results decomposed into the 3 MIXED-PARTS subtasks (common objects, common parts, and unique parts). The decreasing performance across all models delineates the incremental difficulty of the tasks, *i.e*., with common objects being the easiest task, followed by common parts, and finally unique parts as the most challenging task. CALICO consistently outperforms all baselines across all 3 tasks, with strong improvements in scores across all metrics compared to the next best baseline.

E. Limitations

This research introduces CALICO, a model designed for part-focused semantic co-segmentation, incorporating several novel features. While our contributions include the proposal of a valuable new task and the release of a supporting dataset to encourage future research, there are a few limitations and assumptions that warrant discussion. First, CALICO assumes that semantically meaningful correspondences can be established between similar object parts across different images based on visual features. However, this assumption may break down in scenarios where visually similar parts serve different functions. Like many machine learning models, CALICO demonstrates promising results on curated datasets, but its generalizability to complex, real-world environments remains to be fully validated. Future work should address these limitations, potentially through adaptive learning strategies that improve robustness and scalability across diverse, real-world applications.

F. Broader Impact

This work introduces several significant advancements with broad societal implications—both positive and cautionary. The integration of object and part comparison methods holds promise for numerous applications in areas such as automated quality control in manufacturing, enhanced image-based search engines, and more sophisticated systems for digital content management and creation. For example, the proposed new task and models can automate and improve the accuracy of tasks that require detailed visual comparisons, such as quality assurance in manufacturing, potentially increasing efficiency while reducing human error. CALICO offers a more fine-grained understanding of images by identifying and segmenting objects across images based on their constituent common or unique parts. This can greatly aid in fields such as robotics, where agents require a detailed understanding of object parts for precise manipulation or obstacle avoidance tasks. Furthermore, in medical imaging, fine-grained visual decomposition can assist in detailed diagnostic tasks. Improved part-level understanding can also improve accessibility technologies, such as software for the visually impaired, by providing more descriptive and accurate visual summaries.

However, reliance on visual segmentation alone may introduce risks in tasks where non-visual attributes are crucial. Objects with similar appearance but different material properties (*e.g*., plastic *vs*. metal) may be misinterpreted, potentially affecting tasks in high-stakes settings such as surgery or industrial handling of fragile materials. Depending on the application, the integration of other sensory data inputs (tactile, thermal, or acoustic sensors), together with visual data, would be beneficial in mitigating this risk. Future work can also design new datasets and models that consider metadata about material properties and other attributes, and ensure that the model is trained not just to recognize visual object and part similarities/differences, but also to associate these features with the correct properties. Finally, human-in-the-loop solutions that incorporate human oversight can ensure critical decisions are validated,

Task	Method	AP50	mIoU	Recall	SS	S-IoU
Common Objects	Cascade [1, 22, 27]	7.9	37.0	28.2	38.1	29.6
	Multi-Image PartGLEE [32]	1.5	33.5	9.7	87.2	82.8
	Multi-Image VLPart [60]	18.2	42.4	45.6	58.8	55.2
	Multi-Image GLaMM [50]	63.2	71.6	73.3	86.5	86.2
	Multi-Image LISA [27]	60.2	70.0	71.9	86.0	85.3
	CALICO (ours)	**69.2**	**75.2**	**78.5**	**93.7**	**93.4**
Common Parts	Cascade [1, 22, 27]	3.5	18.6	11.7	25.6	6.5
	Multi-Image PartGLEE [32]	1.9	32.0	10.5	**78.4**	63.3
	Multi-Image VLPart [60]	15.7	44.5	29.1	54.4	39.9
	Multi-Image GLaMM [50]	36.7	52.7	48.4	68.5	59.0
	Multi-Image LISA [27]	37.3	54.2	50.5	73.9	63.4
	CALICO (ours)	**38.4**	**56.9**	**51.9**	74.4	**64.4**
Unique Parts	Cascade [1, 22, 27]	5.7	28.1	17.2	32.8	8.2
	Multi-Image PartGLEE [32]	0.1	22.3	8.8	69.9	43.9
	Multi-Image VLPart [60]	6.4	41.5	29.0	64.0	44.4
	Multi-Image GLaMM [50]	28.9	55.3	43.1	75.3	68.4
	Multi-Image LISA [27]	26.6	55.0	44.1	76.2	68.7
	CALICO (ours)	**30.2**	**59.1**	**48.8**	**80.1**	**73.4**
MIXEDPARTS	Cascade [1, 22, 27]	5.7	27.9	19.0	32.2	14.8
	Multi-Image PartGLEE [32]	1.2	29.3	9.7	78.5	63.3
	Multi-Image VLPart [60]	13.4	42.8	34.6	59.1	46.5
	Multi-Image GLaMM [50]	42.9	59.9	54.9	76.8	71.2
	Multi-Image LISA [27]	41.4	59.7	55.5	78.7	72.5
	CALICO (ours)	**45.9**	**63.7**	**59.7**	**82.7**	**77.1**

Table 5. **Per-Task Experimental Results on MIXEDPARTS**. The first three metrics are segmentation-based while the last two are text-based. CALICO surpasses baselines across all three MIXEDPARTS tasks (common objects, common parts, and unique parts).

improving reliability and trustworthiness in deployment.

G. Image Attributions

The three images in Figure 7, used for qualitative illustration of our model's in-context capabilities, are credited respectively to Erik Mclean, Pixabay, and Trace Constant on Pexels. All other images shown in this paper are sourced from publicly available datasets.

Sparkles:

- Extracting all objects from both images:

```
What are the objects in IMAGE#1<Img><ImageHere></Img> and IMAGE#2<Img><ImageHere></Img>?
Reply with only the objects that visible in the images. Only talk about visible features, and limit output to the different objects. Make sure that
each object is covered and described properly. Please talk about both the images and do not repeat objects.
```

- Extracting all parts from both images:

```
What are the parts of objects in IMAGE#1<Img><ImageHere></Img> and IMAGE#2<Img><ImageHere></Img>?
Reply with only the parts of objects that visible in the images. Only talk about visible features, and limit output to the different object and parts.
Make sure that each part is covered and described properly. Please talk about both the images and do not repeat parts.
```

GPT4o:

- Common objects:

```
You will be given description of two images. Understand the description of the images, make a list of the objects in both the images, and identify
common objects. If you do not find any common objects, you can leave it empty.
Return these as a JSON object, strictly in the following format, output nothing else:
{
   'common_objects': ['object1', 'object2', ...]
}
replacing objectX with the actual object names. Or if there are no common objects:
{
   'common_objects': []
}
Here's the description of both the images: <Sparkles' outputs>.
Please maintain the JSON output format with object names for both images.
```

- Common parts:

```
You will be given description of two images. Understand the description of the objects, make a list of the objects and their parts in both the images,
and identify the common parts from all the objects in the images. If you do not find any common parts, you can leave it empty.
Return these as a JSON object, strictly in the following format, output nothing else:
{
   'common_parts': [
      {'image1': [object1, part1], 'image2': [object1, part1]},
      {'image1': [object1, part2], 'image2': [object1, part2]}, ...
      {'image1': [object2, part1], 'image2': [object2, part1]}, ...]
}
replacing objectX and partX with the actual object and part names. Or if there are no common parts:
{
   'common_parts': []
}
Here's the description of both the images: <Sparkles' outputs>.
Please maintain the JSON output format with both object and part names for both images.
```

- Unique parts:

```
You will be given description of two images. Understand the description of the objects, make a list of the objects and their parts in both the images,
and identify the unique parts from all the objects in the image. If you do not find any unique parts, you can leave it empty.
Return these as a JSON object, strictly in the following format, output nothing else:
{
   'unique_parts_image1': [
         ['object1','part1'], ['object1','part2'], ['object2','part1'], ...
      ]
   'unique_parts_image2': [
         ['object1','part1'], ['object1','part2'], ['object2','part1'], ...
      ]
}
replacing objectX and partX with the actual object and part names. Or if there are no unique parts:
{
   'unique_parts_image1': [],
   'unique_parts_image2': []
}
Here's the description of both the images: <Sparkles' outputs>.
Please maintain the JSON output format with both object and part name for both images.
```

LISA:

- Repeat for all common objects:

```
Can you please segment <object> in this image?
```

- Repeat for all (common or unique) parts:

```
Can you please segment <part> of <object> in this image?
```

Figure 10. **Prompts for the Cascaded Pipeline**.

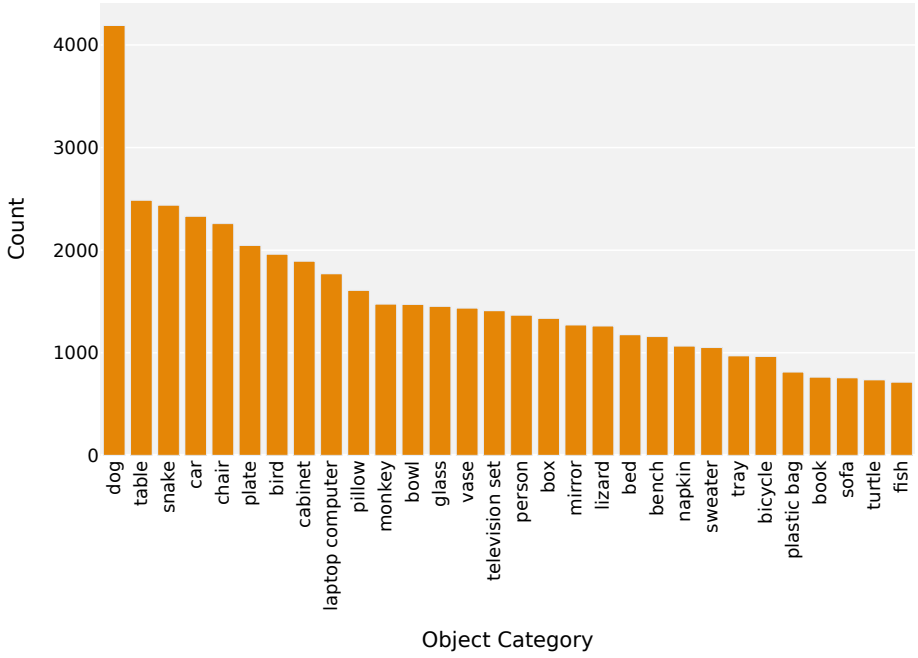

Figure 11. **Top-30 Most Frequent Object Categories in MIXEDPARTS.**

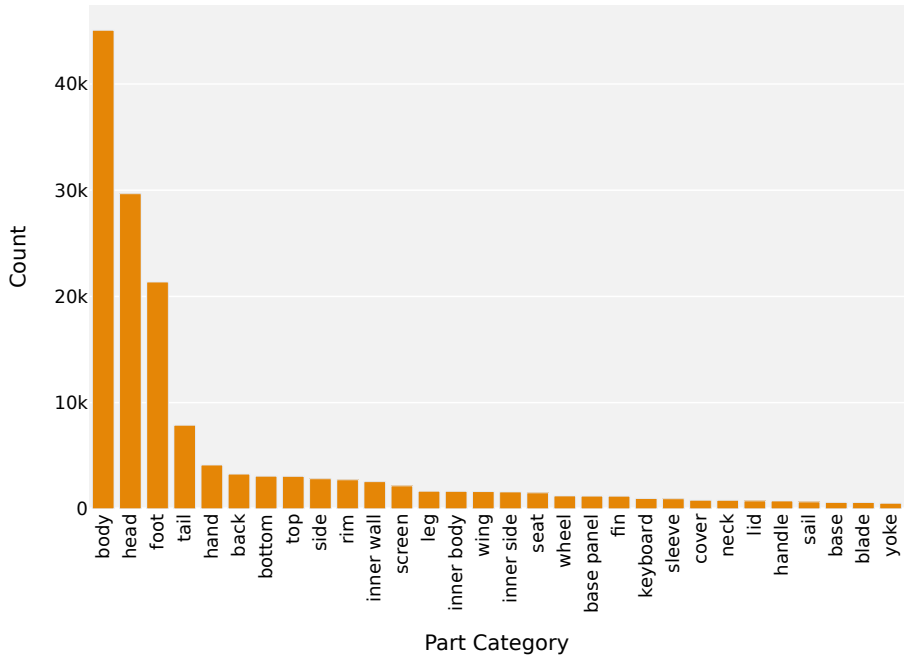

Figure 12. **Top-30 Most Frequent Part Categories in MIXEDPARTS.**

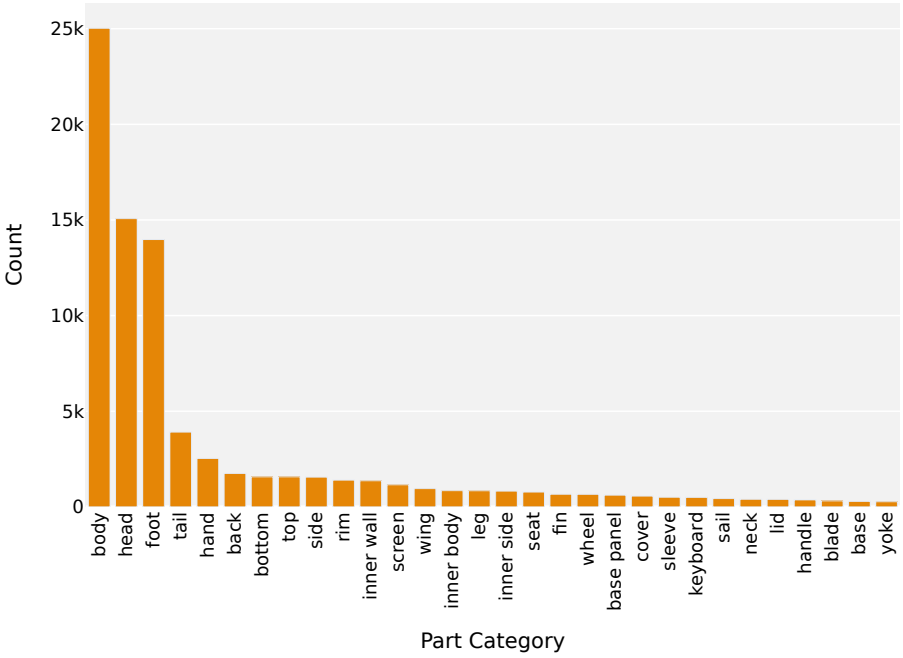

Figure 13. **Top-30 Most Frequent Common Parts in MIXEDPARTS.**

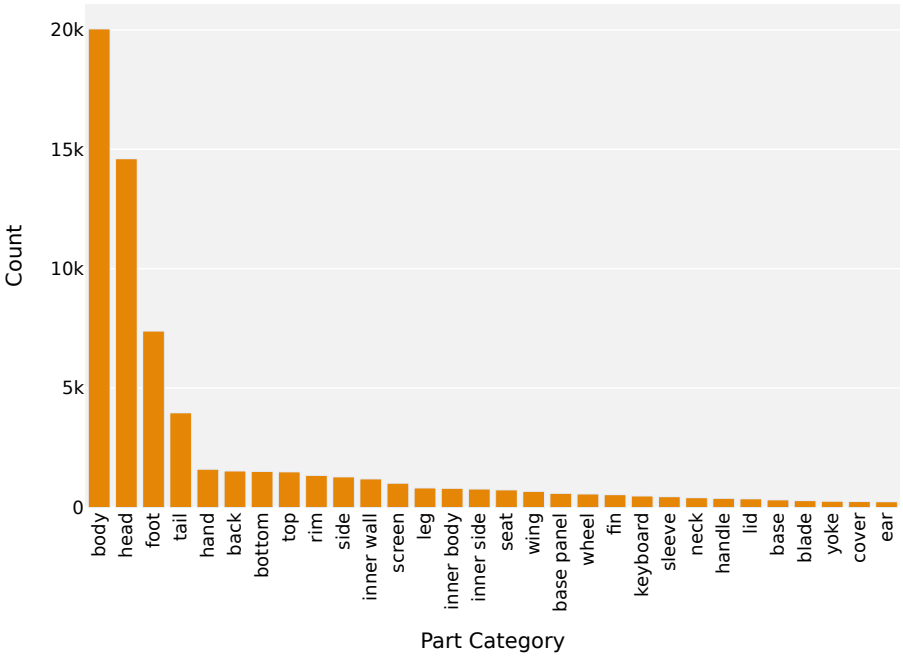

Figure 14. **Top-30 Most Frequent Unique Parts in MIXEDPARTS.**

Dataset	Object	Parts
ADE20K-Part234	airplane	door, fuselage, landing gear, propeller, stabilizer, turbine engine, wing
ADE20K-Part234	armchair	apron, arm, back, back pillow, leg, seat, seat base
ADE20K-Part234	bed	footboard, headboard, leg, side rail
ADE20K-Part234	bench	arm, back, leg, seat
ADE20K-Part234	bookcase	door, drawer, front, side
ADE20K-Part234	bus	bumper, door, headlight, license plate, logo, mirror, wheel, window, wiper
ADE20K-Part234	cabinet	door, drawer, front, shelf, side, skirt, top
ADE20K-Part234	car	bumper, door, headlight, hood, license plate, logo, mirror, wheel, window, wiper
ADE20K-Part234	chair	apron, arm, back, base, leg, seat, skirt, stretcher
ADE20K-Part234	chandelier	arm, bulb, canopy, chain, cord, highlight, light source, shade
ADE20K-Part234	chest of drawers	apron, door, drawer, front, leg
ADE20K-Part234	clock	face, frame
ADE20K-Part234	coffee table	leg, top
ADE20K-Part234	computer	computer case, keyboard, monitor, mouse
ADE20K-Part234	cooking stove	burner, button panel, door, drawer, oven, stove
ADE20K-Part234	desk	apron, door, drawer, leg, shelf, top
ADE20K-Part234	dishwasher	button panel, handle, skirt
ADE20K-Part234	door	door frame, handle, knob, panel
ADE20K-Part234	fan	blade, canopy, tube
ADE20K-Part234	glass	base, bowl, opening, stem
ADE20K-Part234	kitchen island	door, drawer, front, side, top
ADE20K-Part234	lamp	arm, base, canopy, cord, highlight, light source, pipe, shade, tube
ADE20K-Part234	light	aperture, canopy, diffusor, highlight, light source, shade
ADE20K-Part234	microwave	button panel, door, front, side, top, window
ADE20K-Part234	minibike	license plate, mirror, seat, wheel
ADE20K-Part234	ottoman	back, leg, seat
ADE20K-Part234	oven	button panel, door, drawer, top
ADE20K-Part234	person	arm, back, foot, gaze, hand, head, leg, neck, torso
ADE20K-Part234	pool table	bed, leg, pocket
ADE20K-Part234	refrigerator	button panel, door, drawer, side
ADE20K-Part234	sconce	arm, backplate, highlight, light source, shade
ADE20K-Part234	shelf	door, drawer, front, shelf
ADE20K-Part234	sink	bowl, faucet, pedestal, tap, top
ADE20K-Part234	sofa	arm, back, back pillow, leg, seat base, seat cushion, skirt
ADE20K-Part234	stool	leg, seat
ADE20K-Part234	swivel chair	back, base, seat, wheel
ADE20K-Part234	table	apron, drawer, leg, shelf, top, wheel
ADE20K-Part234	television receiver	base, buttons, frame, keys, screen, speaker
ADE20K-Part234	toilet	bowl, cistern, lid
ADE20K-Part234	traffic light	housing, pole
ADE20K-Part234	truck	bumper, door, headlight, license plate, logo, mirror, wheel, window
ADE20K-Part234	van	bumper, door, headlight, license plate, logo, mirror, taillight, wheel, window, wiper
ADE20K-Part234	wardrobe	door, drawer, front, leg, mirror, top
ADE20K-Part234	washer	button panel, door, front, side
PACO-LVIS	basket	base, bottom, cover, handle, inner side, rim, side
PACO-LVIS	belt	bar, buckle, end tip, frame, hole, loop, prong, strap
PACO-LVIS	bench	arm, back, leg, seat, stretcher, table top
PACO-LVIS	bicycle	basket, down tube, fork, gear, handlebar, head tube, pedal, saddle, seat stay, seat tube, stem, top tube, wheel
PACO-LVIS	blender	base, blade, cable, cover, cup, food cup, handle, inner body, seal ring, spout, switch, vapour cover
PACO-LVIS	book	cover, page
PACO-LVIS	bottle	base, body, bottom, cap, capsule, closure, handle, heel, inner body, label, neck, punt, ring, shoulder, sipper, spout, top
PACO-LVIS	bowl	base, body, bottom, inner body, rim
PACO-LVIS	box	bottom, inner side, lid, side
PACO-LVIS	broom	brush, brush cap, handle, lower bristles, ring, shaft
PACO-LVIS	bucket	base, body, bottom, cover, handle, inner body, loop, rim
PACO-LVIS	calculator	body, key
PACO-LVIS	can	base, body, bottom, inner body, lid, pull tab, rim, text
PACO-LVIS	car	antenna, bumper, fender, grille, handle, headlight, hood, logo, mirror, rim, roof, runningboard, seat, sign, splashboard, steeringwheel, taillight, tank, trunk, turnsignal, wheel, window, windowpane, windshield, wiper
PACO-LVIS	carton	bottom, cap, inner side, lid, side, tapering top, text, top
PACO-LVIS	cellular telephone	back cover, bezel, button, screen
PACO-LVIS	chair	apron, arm, back, base, leg, rail, seat, skirt, spindle, stile, stretcher, swivel, wheel
PACO-LVIS	clock	base, cable, case, decoration, finial, hand, pediment
PACO-LVIS	crate	bottom, handle, inner side, lid, side
PACO-LVIS	cup	base, handle, inner body, rim
PACO-LVIS	dog	body, ear, eye, foot, head, leg, neck, nose, tail, teeth
PACO-LVIS	drill	body, handle
PACO-LVIS	drum	base, body, cover, head, inner body, loop, lug, rim
PACO-LVIS	earphone	cable, ear pads, headband, housing, slider
PACO-LVIS	fan	base, blade, bracket, canopy, fan box, light, logo, motor, pedestal column, rod, string
PACO-LVIS	glass	base, body, bottom, inner body, rim
PACO-LVIS	guitar	back, body, bridge, fingerboard, headstock, hole, key, pickguard, side, string
PACO-LVIS	hammer	face, grip, handle, head
PACO-LVIS	handbag	base, body, bottom, handle, inner body, rim, zip
PACO-LVIS	hat	inner side, logo, pom pom, rim, strap, visor
PACO-LVIS	helmet	face shield, inner side, logo, rim, strap, visor
PACO-LVIS	jar	base, body, bottom, cover, handle, inner body, lid, rim, sticker, text
PACO-LVIS	kettle	base, body, cable, handle, inner body, lid, spout, switch
PACO-LVIS	knife	blade, handle
PACO-LVIS	ladder	foot, rail, step, top cap
PACO-LVIS	lamp	base, bulb, cable, finial, pipe, shade, shade cap, shade inner side, switch
PACO-LVIS	laptop computer	back, base panel, cable, camera, keyboard, logo, screen, touchpad
PACO-LVIS	microwave oven	control panel, dial, door handle, inner side, side, time display, top, turntable

Table 6. **Object and Part Taxonomies by Dataset** (to be continued on next page).

Dataset	Object	Parts
PACO-LVIS	mirror	frame
PACO-LVIS	mouse	body, left button, logo, right button, scroll wheel, side button, wire
PACO-LVIS	mug	base, body, bottom, drawing, handle, inner body, rim, text
PACO-LVIS	newspaper	text
PACO-LVIS	pan	base, bottom, handle, inner side, lid, rim, side
PACO-LVIS	pen	barrel, cap, clip, grip, tip
PACO-LVIS	pencil	body, eraser, ferrule, lead
PACO-LVIS	pillow	embroidery
PACO-LVIS	pipe	colied tube, nozzle, nozzle stem
PACO-LVIS	plastic bag	body, handle, hem, inner body, text
PACO-LVIS	plate	base, body, bottom, inner wall, rim
PACO-LVIS	pliers	blade, handle, jaw, joint
PACO-LVIS	remote control	back, button, logo
PACO-LVIS	scarf	body, fringes
PACO-LVIS	scissors	blade, finger hole, handle, screw
PACO-LVIS	screwdriver	handle, shank, tip
PACO-LVIS	shoe	backstay, eyelet, heel, insole, lace, lining, outsole, quarter, throat, toe box, tongue, vamp, welt
PACO-LVIS	slipper	insole, lining, outsole, strap, toe box, vamp
PACO-LVIS	soap	base, body, bottom, cap, capsule, closure, handle, label, neck, punt, push pull cap, ring, shoulder, sipper, spout, top
PACO-LVIS	sponge	rough surface
PACO-LVIS	spoon	bowl, handle, neck, tip
PACO-LVIS	stool	footrest, leg, seat, step
PACO-LVIS	sweater	body, cuff, hem, neckband, shoulder, sleeve, yoke
PACO-LVIS	table	apron, drawer, inner body, leg, rim, shelf, stretcher, top, wheel
PACO-LVIS	tape	roll
PACO-LVIS	telephone	back cover, bezel, button, screen
PACO-LVIS	television set	base, bottom, button, side, top
PACO-LVIS	tissue paper	roll
PACO-LVIS	towel	body, border, hem, terry bar
PACO-LVIS	trash can	body, bottom, hole, inner body, label, lid, pedal, rim, wheel
PACO-LVIS	tray	base, bottom, inner side, inner wall, outer side, rim
PACO-LVIS	vase	body, foot, handle, mouth, neck
PACO-LVIS	wallet	flap, inner body
PACO-LVIS	watch	buckle, case, dial, hand, lug, strap, window
PACO-LVIS	wrench	handle, head
PartImageNet	airplane (aeroplane)	body, engine, head, tail, wing
PartImageNet	alligator (reptile)	body, foot, head, tail
PartImageNet	antelope (quadruped)	body, foot, head, tail
PartImageNet	ape (biped)	body, foot, hand, head, tail
PartImageNet	badger (quadruped)	body, foot, head, tail
PartImageNet	bear (quadruped)	body, foot, head, tail
PartImageNet	bird (bird)	body, foot, head, tail, wing
PartImageNet	boat (boat)	body, sail
PartImageNet	camel (quadruped)	body, foot, head, tail
PartImageNet	cat (quadruped)	body, foot, head, tail
PartImageNet	cheetah (quadruped)	body, foot, head, tail
PartImageNet	cougar (quadruped)	body, foot, head, tail
PartImageNet	crocodile (reptile)	body, foot, head, tail
PartImageNet	dog (quadruped)	body, foot, head, tail
PartImageNet	fish (fish)	body, fin, head, tail
PartImageNet	fox (quadruped)	body, foot, head, tail
PartImageNet	frog (reptile)	body, foot, head, tail
PartImageNet	goat (quadruped)	body, foot, head, tail
PartImageNet	leopard (quadruped)	body, foot, head, tail
PartImageNet	lizard (reptile)	body, foot, head, tail
PartImageNet	mink (quadruped)	body, foot, head, tail
PartImageNet	monkey (biped)	body, foot, hand, head, tail
PartImageNet	otter (quadruped)	body, foot, head, tail
PartImageNet	ox (quadruped)	body, foot, head, tail
PartImageNet	panda (quadruped)	body, foot, head, tail
PartImageNet	polecat (quadruped)	body, foot, head, tail
PartImageNet	shark (fish)	body, fin, head, tail
PartImageNet	sheep (quadruped)	body, foot, head, tail
PartImageNet	snake (snake)	body, head
PartImageNet	squirrel (quadruped)	body, foot, head, tail
PartImageNet	swine (quadruped)	body, foot, head, tail
PartImageNet	tiger (quadruped)	body, foot, head, tail
PartImageNet	turtle (reptile)	body, foot, head, tail
PartImageNet	water buffalo (quadruped)	body, foot, head, tail
PartImageNet	weasel (quadruped)	body, foot, head, tail
PartImageNet	whale (fish)	body, fin, head, tail
PartImageNet	wolf (quadruped)	body, foot, head, tail

Table 6. **Object and Part Taxonomies by Dataset** (continued).