

A. Overview

We give an overview over the contents of the Appendix.

- In Appendix B, we present additional details about the creation of initial LLM queries for DASH-LLM, including the prompts used to create the queries with Llama.
- In Appendix C, we break down the optimization of the DASH-OPT image queries in the latent space of the distilled SDXL model.
- In Appendix D, we give additional details about the ReLAION exploration/exploitation retrieval.
- In Appendix E, we investigate the influence of an object’s occurrence frequency on hallucination rates.
- In Appendix F, we present additional results for DASH on PaliGemma, LLaVA-NeXT Vicuna and Mistral. In Appendix F.3, we also present DASH results on ReLAION next to the most similar images from COCO and Objects365.
- In Appendix G, we further explore the performance of the object detector in DASH to filter out images containing the object.
- In Appendix H, we show examples of COCO annotation errors and discuss their effect on the POPE benchmark.
- In Appendix I, we present extended results about the transfer between DASH images to other VLMs and also present information about all VLMs used in the paper in Appendix I.1 and the true positive rate calculation in Appendix I.2
- In Appendix J, we describe the image selection process and discuss different metrics for our proposed benchmark DASH-B. Additionally, we report more results on a range of VLMs.
- In Appendix K, we give further details about the mitigation finetuning using DASH.
- In Appendix L, we provide a proof of concept for a possible application of our pipeline to the reverse task and discuss problems.
- In Appendix M, we examine the generalization of DASH results to different prompts than the one used in our experiments.

B. DASH-LLM Prompt

The prompts supplied to Llama-3.1-70B [9] to create the queries for DASH-LLM are given in Fig. 7 and 8. We also use the same queries to initialize the generation of the image queries in DASH-OPT. To generate the queries, we use the system prompt provided in Figure 7. We then pass the input “object: OBJ” to the LLM, which generates an initial list of 50 queries. Since we noticed that these initial queries can sometimes contain references to the object or duplicates, we use a simplified version of chain-of-thought prompting [55]. After the LLM generates the initial list of 50 queries, we pass the follow-up prompt provided in Fig-

ure 8 to the model, which responds with an updated list of 50 queries.

C. DASH-OPT Optimization

Optimization examples: In Fig. 9, we present the optimization trajectory of DASH-OPT for two images. In Fig. 10, we provide additional examples where we show only the initialization (*i.e.*, the image generated by SDXL using the text query from DASH-LLM without any optimization) and the final query image produced by DASH-OPT after optimization. These examples illustrate that DASH-OPT is capable of generating unexpected FP-hallucinations. For instance, it introduces beads for “leopard,” which are absent from the original caption that merely describes “a rock’s cracks and fissures”. Similarly, we demonstrate the transformation of “a set of scented lotions ... on a shelf” into a scene of a person shopping for “bathing bombs” when optimizing for hallucinations related to the object “bathtub”.

The corresponding retrieved images, which validate that these phenomena are not limited to synthetic data but also occur with real images, can be found in Fig. 3 for “leopard” and Fig. 13 for “bathtub”.

Implementation details: For DASH-OPT, we use the distilled version of Stable Diffusion XL (SDXL) [37] from [40]. In particular, we use the single-step SDXL U-Net together with the Latent Consistency Model (LCM) scheduler [31], setting the start timestep to 800.

For optimization, we use the Adam optimizer [19] for 25 steps with a step size of 0.1, applying a linear warmup over the first 3 steps. The gradient is clipped to an L_2 norm of 0.1 at every step. When using a deterministic scheduler like the single-step LCM scheduler, three variables determine the output of the diffusion process. The first is the Gaussian random latent drawn at the start of the generation. The second and third are the text encodings of the user prompt generated by the two different CLIP text encoders in SDXL. We optimize all three variables and additionally apply a step size factor of 0.1 for the random latent. For the random latent, we also employ the chi-square latent regularization method from [42]. Note that the text encodings are initialized using the text queries from DASH-LLM (Appendix B). As optimization loss, we use Eq. (3). Note that $p_{\text{det}}(\text{OBJ} \mid q(C))$ is computed as the maximum confidence overall bounding boxes and thresholded to 0 for detection probabilities smaller than 0.05. Since the optimization problem is highly non-convex, the last image is not necessarily the one with the best overall loss, and hence, we use the one with the lowest loss over all generated images as the query for DASH-OPT.

Different models use different tokens corresponding to “yes” (e.g. “yes”, “Yes”, “Yes.”) during evaluation. Thus, for each VLM, we choose the target token for the optimiza-

tion accordingly.

The optimization takes around 50 seconds for PaliGemma and one minute for the LLaVA-NeXT models on an NVIDIA A100 GPU with 40GB of memory.

D. Retrieval process, exploration and exploitation

The ReLAION-5B [39, 44] index, which we use for retrieval during the exploration and exploitation stages, is based on OpenCLIP ViT-H [15]. During retrieval, we apply DreamSim [11] to remove near-duplicate images with a similarity score greater than 0.9, as LAION is estimated to contain up to 30% duplicated data [54]. For clustering in the exploitation phase, we first group all images retrieved for the same image during the exploration phase into pre-clusters. These pre-clusters are then merged using agglomerative clustering to form the final clusters. We employ average linkage based on DreamSim distances, with a maximum allowed merge threshold of 0.6.

E. Impact of object occurrence frequency on object hallucinations

We run DASH on subsets of OpenImages with different occurrence frequencies and show the average number of images per object for each split found by DASH for PaliGemma LLaVA-NeXT Vicuna, and LLaVA-NeXT Mistral in Fig. 11. The results for PaliGemma are particularly interesting, as the model was trained on a similar task (“Is there a *object* in the image?”) on this dataset. Overall, it is easier to find systematic hallucinations for objects that are very rare (on average 506 images) and gets harder if they occur more frequently. Especially for the frequent objects, the optimized queries help to find more of the rarer hallucinations, resulting in significantly more images per object for DASH-OPT on the corresponding splits compared to DASH-LLM. The observed trends for PaliGemma also hold true for LLaVA-NeXT Vicuna and LLaVA-NeXT Mistral. Both are much more vulnerable on rare objects, and object frequency seems to be a strong indicator of an object’s vulnerability although the LLaVA-NeXT models are not trained on OpenImages. However, it is possible that the distribution of images in OpenImages is similar to that of other large-scale datasets, such as those used to train the CLIP [38] models employed in LLaVA-NeXT.

F. DASH Results Extended

F.1. Additional qualitative examples

In Figures 12 to 17 we present additional retrieval results, similar to those from Fig. 3 for DASH-LLM and DASH-OPT. In particular, we include results for LLaVA-NeXT Vicuna and Mistral. As these Figures demonstrate, all 3 VLMs

suffer from a substantial amount of type II hallucinations.

In Fig. 12, we show the clusters of images generated using DASH-LLM for PaliGemma. The examples illustrate how the LLM-generated queries lead to images that are logically connected to the object in a semantic sense. For instance, for the object “Barracouta,” we observe coastal towns and harbors built in Minecraft, likely reflecting the object’s marine context. For “Fireboat,” we find images of the police using water cannons, which are often commonly found on a “Fireboat”. In Fig. 13, we present examples of clusters identified using DASH-OPT for the same VLM and various objects, highlighting cases of “unknown unknowns”. For instance, for the object “Bathtub,” the cluster includes colorful images of bath bombs, rather than bathtubs, suggesting that the model has learned to associate the object label with related items rather than the physical object itself. Similarly, for the object “Puck,” instead of hockey pucks, the cluster prominently features images of stacked oranges and other spherical objects, reflecting a semantic confusion between shape and context. For “Sulphur Butterfly,” the cluster contains ornamental decorations and holiday-themed items, diverging significantly from the actual insect.

The clusters shown in Fig. 14 illustrate examples generated using DASH-LLM with LLaVA-NeXT Vicuna as the VLM. These examples reflect expected yet interesting associations generated by the model based on LLM-guided queries. For instance, for the object “Academic Gown,” the cluster includes university seals and architectural elements from academic institutions, which are logically associated with the concept of academia but deviate visually from the actual object. Similarly, for “Chain Mail,” the model identifies medieval swords and weaponry, which are contextually related to chain mail in historical settings. The object “Fountain Pen” generates a cluster dominated by handwritten scripts and paper stacks, reinforcing a semantic association with writing and stationery. The cluster for “Coral Fungus” features lichen-covered tree bark and textures, highlighting a broader misinterpretation of the object’s actual form and an emphasis on natural growth patterns. Finally, for “Postcard,” the cluster predominantly displays boardwalks and scenic ocean views, which align with common themes of postcards.

In Fig. 15, we showcase clusters generated using DASH-OPT with LLaVA-NeXT Vicuna, highlighting more unexpected results where the VLM demonstrates surprising or unintended associations. For “Dogsled,” the cluster contains images of snowshoes and other winter-related gear, which are contextually linked to snowy environments but do not represent the object itself. The object “Strawberry” leads to a cluster featuring images of festive door decorations and wreaths. This unexpected association likely arises from the model’s inability to separate the red and green

color palette of strawberries from decorative elements. For “Beehive,” the cluster includes a surprising array of human portraits, particularly women in colorful settings. This suggests that the VLM may associate the term “beehive” with a hairstyle rather than the physical structure created by bees.

In Fig. 16, we present examples of clusters generated using DASH-LLM with LLaVA-NeXT Mistral. For the object “Band Aid,” the cluster contains images of people holding or bandaging injured wrists, reflecting a logical semantic association with the concept of injury and care. Similarly, for “Gondola,” the cluster features small shops, which aligns with a broader cultural and contextual understanding of gondolas as part of scenic, tourist-driven environments. The object “Dumbbell” leads to a cluster of colorful exercise balls, emphasizing fitness and gym-related settings, likely derived from contextual overlaps. For “Lighter,” the cluster showcases dimly lit rooms with a smoky haze in video games, reflecting a plausible connection to the object’s typical use in dark settings. The “Lighthouse” cluster includes solitary piers and fishing-related environments, reinforcing the model’s interpretation of the lighthouse’s association with remote coastal locations.

In Fig. 17, we present clusters generated using DASH-OPT with LLaVA-NeXT Mistral. For the object “Agama,” the cluster prominently features various wild cats. For “Bulletproof Vest,” the cluster includes images of surveillance and monitoring rooms with large screens, likely due to the association of vests with security and law enforcement. The object “Horizontal Bar” leads to a cluster filled with water bottles and similar cylindrical objects, reflecting a superficial visual similarity in shape but entirely unrelated semantics. For “Shallot,” the cluster displays images of modern kitchens and industrial food preparation areas, suggesting that the VLM has learned to associate the object with its culinary context rather than its specific visual characteristics. For “Bluehead,” instead of the fish species, the cluster includes images of blue-themed furniture and interior designs, driven by the color association rather than the object itself. For “Bird,” the cluster prominently features butterflies and flowers, illustrating a misalignment between the object category and the broader semantic associations of natural imagery. Lastly, the clusters for “Hat” and “Balloon” show out-of-distribution images that are not logically connected to the object.

F.2. All Clusters Visualizations

In Fig. 20, we present all clusters identified for DASH-LLM and DASH-OPT for the object “Ptarmigan.” While “Ptarmigan” refers to a bird species, the clusters reveal a range of false positives, including images of mountain landscapes, alpine environments, abstract artistic representations, and completely unrelated objects. This indicates that the VLM conflates semantic and contextual cues with visual content,

leading to systematic hallucinations.

Interestingly, many of these errors may stem from the existence of places named “Ptarmigan,” such as “Ptarmigan Peak” in Colorado, Utah, and Alaska, or “Ptarmigan Ridge” and “Ptarmigan Traverse” in Washington. Even though these locations are unrelated to the bird, the VLM erroneously associates them with the object “Ptarmigan.” Our analysis confirms that these places are distinct mountainsides with unrelated names, demonstrating that the VLM has learned a flawed representation of “Ptarmigan” that includes a variety of unrelated mountainous scenes.

Additionally, DASH-OPT uncovers further “unknown unknowns,” such as rare or abstract scenes where a ptarmigan is highly unlikely, including auroras, surreal artwork, and stylized objects.

In Fig. 21, we present all clusters found for DASH-LLM and DASH-OPT for the object “Baumkuchen” on LLaVA-NeXT Mistral. For DASH-LLM, we observe that the clusters include no images of Baumkuchen, a traditional German layered cake, but a variety of unrelated objects and scenes. These false positives encompass German cultural artifacts, traditional buildings, festivals, and abstract artistic representations, indicating that the VLM has conflated “Baumkuchen” with broader semantic or cultural cues tied to German traditions.

DASH-OPT uncovers additional “unknown unknowns.” Alongside unrelated cultural goods like Christmas decorations, traditional crafts, and books which we have also found for DASH-LLM, we also find additional systematic vulnerabilities. For example, we find a cluster of 111 images containing fountain pens, but also a cluster of 66 images containing wooden kitchen utensils. We also note that the cluster of size 8 which contains cake does not contain any images of “Baumkuchen”.

F.3. DASH vs Reference Datasets

As stated in the main paper, we also compare our DASH images to reference datasets such as COCO or Objects365 which are commonly used to construct hallucination benchmarks. In Fig. 18, we demonstrate images that cause PaliGemma to detect the target class, identified using DASH-OPT, alongside their nearest neighbors from the reference datasets COCO and Objects365. We observe that neither the full COCO training set (80K samples) nor Objects365 (1.7M samples) contain the systematic errors uncovered by DASH, as all nearest neighbors are not detected by the VLM. This highlights that our open-world search in ReLAION-5B is necessary to detect these hallucinations, which would not be possible even with reasonably large datasets like Objects365. Specifically, with DASH, we find that PaliGemma incorrectly answers “yes” for colorful “Wellington boots” as “apple” and for “Baobab trees” as “sausage.”

These examples illustrate the limitations of relying solely on existing datasets for identifying hallucinations in VLMs. In particular, just because a target object is contained in a dataset like COCO or Objects365, as are all examples presented in Fig. 18, does not guarantee that objects that are not contained in this dataset cannot cause a VLM to hallucinate the target object. Our method uncovers novel failure cases that are absent in standard benchmarks, emphasizing the importance of an open-world search strategy for comprehensive evaluation.

F.4. Larger exploration range of DASH-OPT over DASH-LLM

In Fig. 19, we present extended version of Fig. 4 for PaliGemma, LLaVA-NeXT Vicuna, and LLaVA-NeXT Mistral which demonstrates that DASH-OPT achieves a greater diversity of images than DASH-LLM.

G. Object Detector: False Negative Rate vs False Positive Rate

For the object detector OWLv2 [33] in our pipeline, we pass the object name OBJ and the image to the model. The model then returns a predefined number of bounding boxes, each with a confidence score in the range [0,1]. We take the maximum confidence over all bounding boxes and use this as the probability of the image containing the object, *i.e.*, $p_{\text{det}}(\text{OBJ} \mid \text{img})$. We then reject all images where p_{det} is greater than our threshold of 0.1.

To verify our automatic pipeline, and especially the conservative threshold for the object detector, we manually labeled 10 random images for each object for DASH-OPT on PaliGemma. As stated in Sec. 4.1, we use the labels “yes” if the object is visible, “no” if it is absent, and “ambiguous” for corner cases. Across all images, we find that 5.2% contain the object and 7.8% are ambiguous.

We additionally provide a per-dataset breakdown over object classes in Fig. 22, where we plot the “yes,” “no,” and “ambiguous” ratios. Notably, most objects do not contain any instances of the specified object. Instead, the majority of errors stem from a few object categories where the object detector exhibits systematic issues. Qualitative examples are shown in Fig. 23 and Fig. 24.

We observed that some images were labeled as “ambiguous” in our human evaluation due to various factors. For instance, in the cases of “Barn” and “Bookshop,” the limited image resolution made it difficult to identify specific objects; distinguishing a house from a barn in an aerial view is nearly impossible. For “Kai yang,” a Thai chicken dish, while the depicted dishes might contain chicken, it is challenging to determine whether they are specifically “Kai yang.” Interestingly, a reverse image search labels the first image as “kebab.”

Similarly, for “Cowry,” which refers to small sea snails, even if our human labelers could not identify any, it is difficult to guarantee their absence in the image. In the case of “Airplane,” the interiors shown could represent futuristic airplane or train designs, making it ambiguous. For “Train,” the low image resolution hinders the inference of specific objects’ presence.

Furthermore, ambiguity arises from the object labels themselves in some datasets. For example, in Objects365, “Glasses” refers to eyewear, but the images often contain multiple glass objects, causing confusion. Likewise, “Soccer” refers exclusively to a soccer ball in Objects365, whereas the sport itself is not a well-defined object, leading to uncertainty about whether to label images of referees as “yes” or “no.”

In addition to ambiguous cases, we identified several failure cases of the object detector during our human evaluation. All images in these cases had a confidence score below the threshold of 0.1 and were therefore not rejected by our automated pipeline. For “Mountain bike,” the primary issue was that the objects were very small and difficult to spot. In other instances, such as “Pot” or “Faucet,” the objects are clearly visible, but the object detector failed to recognize them. For “Car,” the detector did not classify trucks or vans as cars. Similarly, for “Mouse” and “Egg,” the detector struggled with distribution shifts, failing to recognize comic or plush mice and colored eggs, respectively.

These observations suggest that while our object detector generally performs well, there are specific categories and scenarios where it struggles, either due to ambiguity in object definitions or limitations in detecting certain object variations.

H. Effect of COCO annotation errors on POPE

Current VLMs only produce a small number of false positives on the POPE benchmark, *e.g.* PaliGemma predicts “yes” on 137 out of the 4500 samples which do not contain the corresponding object according to the COCO annotations. We re-annotate these images and assign the labels

- “yes” if the object is visible in the image,
- “no” if the object is **not** visible in the image,
- “ambiguous” for corner cases where it is not clear whether the object is present or not.

The result of our labeling is that 35 (25.5%) of the alleged false positives actually do contain the object which means that the model reply “yes” is actually correct (see Fig. 25 for examples). In addition, 31 (22.6%) of the images receive the label “ambiguous”. This large amount of label noise among the remaining false positives indicates that the POPE benchmark is saturated.

```
1 As an AI language model assistant, your task is to provide descriptive captions for images showing
   spurious features.
2
3 A spurious feature is a visual element that frequently co-occurs with a given object in images and may
   cause AI models to incorrectly recognize the object, even when it is not present.
4
5 Task Overview:
6
7 You will be given:
8 - An object.
9
10 Your job is to:
11
12 1. Think of potential spurious features: Identify objects, scenes, or elements that frequently co-occur
   with the given object in images. These should not include any parts or components of the object
   itself.
13
14 2. Generate 50 unique and diverse prompts describing images that contain only these spurious features,
   without including the object itself or any of its parts.
15
16 Important Guidelines:
17
18 - Do Not Mention the Object Name or Any Part of It: Avoid any direct or indirect references to the
   object name. If the object name is a composite or compound word, do not include any part of the
   object name in the prompts. For example, if the object is "firetruck," do not use "fire" or "truck"
   in the prompts.
19
20 - Do Not Mention Parts of the Object: Do not include any parts or components of the object in the
   prompts. For example, if the object is "mountainbike," do not use "handlebar," "gear shift," or "
   saddle" in the prompts.
21
22 - Do Not Include the Object Name in Written Text: Do not create prompts that refer to written text
   containing the object name or any part of it. For example, avoid descriptions like "a sign that
   says 'hummingbird'."
23
24 - Focus on Spurious Features: Use features that are likely correlated with the object due to frequent
   co-occurrence in images.
25
26 - Combining Elements: You may combine elements if they logically make sense to appear together in one
   image. Do not combine elements unlikely to co-occur.
27
28 - Ensure Diversity: Each prompt should be unique and cover different aspects of the spurious features.
29
30 - Avoid Repetition: Do not repeat prompts or make minor variations of the same prompt.
31
32 - Style and Detail: Write clear, creative, and descriptive prompts. Keep each prompt concise.
33
34 - Language and Grammar: Use proper grammar and spelling.
35
36 - Content Restrictions: Do not include offensive, sensitive, or inappropriate content.
37
38 - Avoid Bias: Ensure prompts are inclusive and free from cultural, gender, or racial bias.
39
40 - Verification: Before submitting, review the prompts to ensure they comply with all guidelines.
```

Figure 7. DASH-LLM prompt for generating the text queries (1/3)

```

1
2 Examples:
3
4 For the object "hummingbird":
5
6 - Correct Prompts:
7   - "Close-up of a bird feeder hanging in a lush garden."
8   - "A garden filled with vibrant red flowers."
9   - "Green foliage glistening after a rainfall."
10  - "A bird feeder surrounded by blooming plants."
11  - "Red tubular flowers swaying in the breeze."
12
13 - Incorrect Prompts (Do Not Use):
14   - "A hummingbird hovering near a flower."
15   - "Close-up of a hummingbird's wings in motion."
16   - "A small bird with iridescent feathers perched on a branch."
17   - "A sign with the word 'hummingbird' in a botanical garden."
18
19 For the object "firetruck":
20
21 - Correct Prompts:
22   - "A fire station with bright red doors."
23   - "Close-up of a spinning emergency siren light."
24   - "Firefighters conducting a training drill."
25   - "A tall ladder reaching up the side of a building."
26   - "Protective gear hanging neatly in a station locker room."
27
28 - Incorrect Prompts (Do Not Use):
29   - "A bright red firetruck parked on the street."
30   - "Children waving at a passing firetruck."
31   - "A sign that reads 'Fire Station No. 1'."
32   - "A red truck with emergency equipment."
33   - Using the words "fire" or "truck" in the prompts.
34
35 For the object "mountainbike":
36
37 - Correct Prompts:
38   - "A winding trail cutting through a dense forest."
39   - "A helmet resting on a tree stump beside a path."
40   - "Sunlight filtering through trees along a forest trail."
41   - "A backpack leaning against a wooden signpost on a hillside."
42   - "A group of friends hiking through mountainous terrain."
43
44 - Incorrect Prompts (Do Not Use):
45   - "A mountainbike leaning against a tree."
46   - "Close-up of a mountainbike's gears."
47   - "A cyclist adjusting the saddle of a mountainbike."
48   - "A sign that says 'Mountainbike Trail Ahead'."
49   - Using the words "mountain" or "bike" in the prompts.
50   - Mentioning parts like "handlebar," "gear shift," or "saddle."

```

Figure 7. DASH-LLM prompt for generating the text queries (2/3)


```

1
2 Formatting Instructions:
3
4 - Start each prompt on a new line, numbered sequentially from 1 to 50.
5
6 - The format should be:
7
8     1: <prompt_1>
9     2: <prompt_2>
10    3: <prompt_3>
11    ...
12    50: <prompt_50>
13
14 User Input Format:
15
16 The user will provide the object in the following format:
17
18 object: <object name>
19
20 Your Response:
21
22 - Return exactly 50 prompts per user request.
23
24 - Ensure that the last line of your response starts with:
25
26     50: <prompt_50>
27
28 - Under no circumstances should you include any content in your response other than the 50 prompts. Do
    not include explanations, apologies, or any additional text.
29
30 Summary:
31
32 - Do not mention the object name or any part of it. If the object name is a composite or compound word,
    do not include any part of it in the prompts.
33
34 - Do not mention parts or components of the object.
35
36 - Do not create prompts that refer to written text containing the object name or any part of it.
37
38 - Focus on spurious features that frequently co-occur with the object.
39
40 - You may combine elements if they logically co-occur in an image.
41
42 - Ensure diversity and uniqueness in the prompts.
43
44 - Use proper language and avoid any inappropriate content.
45
46 - Review all prompts for compliance before submitting.
47
48 - Under no circumstances should you include any content in your response other than the 50 prompts. Do
    not include explanations, apologies, or any additional text.
49
50 Remember, the goal is to create prompts that could lead an AI model to falsely recognize the object due
    to the presence of spurious features, even though the object itself is not present in the images.

```

Figure 7. DASH-LLM prompt for generating the text queries (3/3)

```

1 Please review the list of prompts you previously generated and check for any mistakes or deviations
  from the guidelines. Identify any prompts that do not fully comply with the instructions. Then,
  generate a new list of 50 prompts that strictly adhere to all the guidelines provided.
2
3 Important Guidelines:
4
5 - Do not mention the object name or any part of it. If the object name is a composite or compound word,
  do not include any part of the object name in the prompts.
6 - Do not mention parts or components of the object.
7 - Do not create prompts that refer to written text containing the object name or any part of it.
8 - Focus on spurious features that frequently co-occur with the object.
9 - You may combine elements if they logically co-occur in an image.
10 - Ensure diversity and uniqueness in the prompts.
11 - Use proper language and avoid any inappropriate content.
12 - Review all prompts for compliance before submitting.
13 - Under no circumstances should you include any content in your response other than the 50 prompts. Do
  not include explanations, apologies, or any additional text.
14
15 Formatting Instructions:
16
17 - Start each prompt on a new line, numbered sequentially from 1 to 50.
18 - The format should be:
19
20 1: <prompt_1>
21 2: <prompt_2>
22 3: <prompt_3>
23 ...
24 50: <prompt_50>
25
26 - Ensure that the last line of your response starts with:
27
28 50: <prompt_50>
29
30 Remember, your goal is to create prompts that could lead an AI model to falsely recognize the object
  due to the presence of spurious features, even though the object itself is not present in the
  images.
31
32 Now, generate the corrected list of 50 prompts.

```

Figure 8. DASH-LLM follow-up prompt for generating the text queries










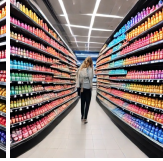


Initialization	Step 5	Step 10	Step 15	Step 20	Step 25
"Leopard" - Prompt: "A close-up of a rock's cracks and fissures."					
VLM: "no"	VLM: "no"	VLM: "no"	VLM: "yes"	VLM: "yes"	VLM: "yes"
$p_{\text{yes}} : 0.04$	$p_{\text{yes}} : 0.05$	$p_{\text{yes}} : 0.06$	$p_{\text{yes}} : 0.75$	$p_{\text{yes}} : 0.77$	$p_{\text{yes}} : \mathbf{0.79}$
$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$
					
"Bathtub" - Prompt: "A set of scented lotions arranged on a shelf."					
VLM: "no"	VLM: "no"	VLM: "no"	VLM: "no"	VLM: "yes"	VLM: "yes"
$p_{\text{yes}} : 0.03$	$p_{\text{yes}} : 0.06$	$p_{\text{yes}} : 0.04$	$p_{\text{yes}} : 0.08$	$p_{\text{yes}} : \mathbf{0.71}$	$p_{\text{yes}} : 0.47$
$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$
					

Figure 9. Optimization trajectories for DASH-OPT for PaliGemma. For each example, we present the object label, the DASH-LLM query used to initialize the generation, as well as the answer and "yes" probability from the VLM and the probability from the detector. Through our optimization process, we can uncover model-specific "unknown unknowns," such as the "beads" (see Fig. 3 for retrieved images) or the "bath bombs" (see Fig. 13). Since the last image is not necessarily the best, we select the image with the lowest loss as the query.





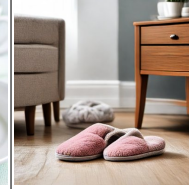







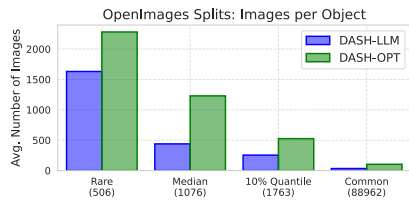
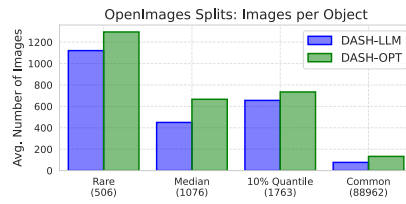
Initialization	DASH-OPT	Initialization	DASH-OPT	Initialization	DASH-OPT
PaliGemma - "Gar" "A water's edge with a few rocks and pebbles."		PaliGemma - "Cabbage Butterfly" "A white flower with a yellow center and a delicate, lacy texture."		LLaVA-NeXT Vicuna - "Pill bottle" "A pair of slippers next to a piece of furniture."	
VLM: "no"	VLM: "yes"	VLM: "no"	VLM: "yes"	VLM: "no"	VLM: "yes"
$p_{\text{yes}} : 0.03$	$p_{\text{yes}} : 0.68$	$p_{\text{yes}} : 0.01$	$p_{\text{yes}} : 0.63$	$p_{\text{yes}} : 0.21$	$p_{\text{yes}} : 0.87$
$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$
					
LLaVA-NeXT Vicuna - "Gondola" "A beautiful Murano glass vase on display in a shop window."		LLaVA-NeXT Mistral "Beehive" "A person holding a frame in a field of blooming flowers."		LLaVA-NeXT Mistral "Fortune Cookie" "A vibrant street festival with dragon dancers."	
VLM: "no"	VLM: "yes"	VLM: "no"	VLM: "yes"	VLM: "no"	VLM: "yes"
$p_{\text{yes}} : 0.15$	$p_{\text{yes}} : 0.61$	$p_{\text{yes}} : 0.03$	$p_{\text{yes}} : 0.52$	$p_{\text{yes}} : 0.18$	$p_{\text{yes}} : 0.85$
$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.00$	$p_{\text{det}} : 0.07$	$p_{\text{det}} : 0.00$
					

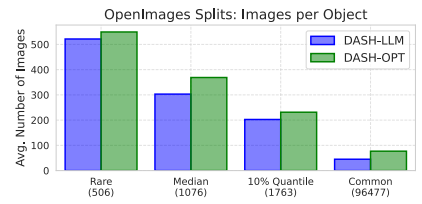
Figure 10. We show examples of DASH-OPT query images after optimization together with the initialization generated from the text query. Our optimization is able to generate images that make VLM hallucinate from non-successful prompts without generating the object.



(a) PaliGemma



(b) LLaVA-NeXT Vicuna



(c) LLaVA-NeXT Mistral

Figure 11. **Influence of object frequencies:** Histogram showing the average number of success images per object category across the OpenImages splits for DASH-LLM and OPT with PaliGemma, LN Vicuna and LN Mistral. The average number of training examples per class in the full 9M OpenImages dataset is indicated in parentheses. The plot reveals that rarer concepts are more susceptible to FP-hallucinations, whereas common concepts with tens of thousands of examples are much less prone to such errors.

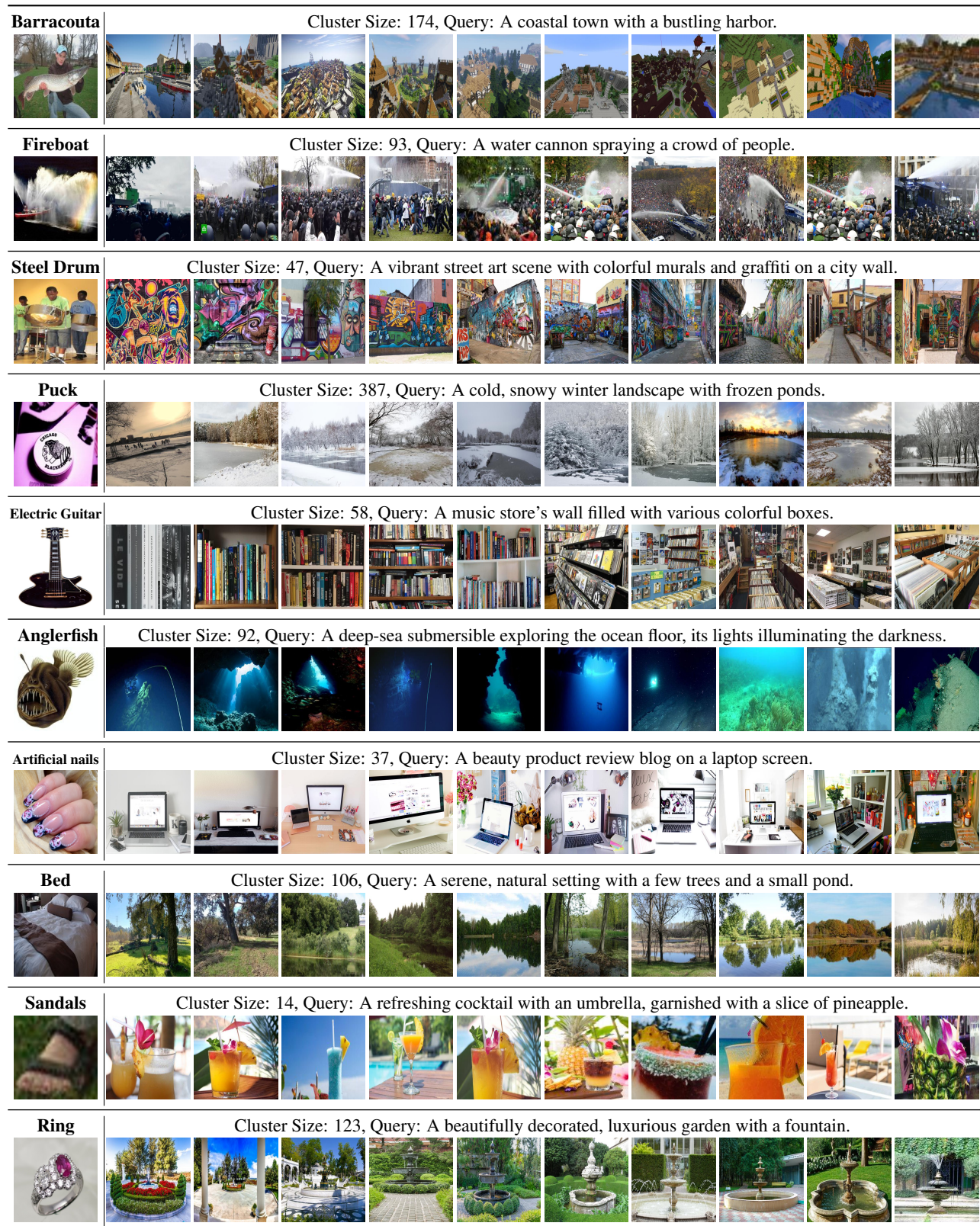


Figure 12. DASH-LLM PaliGemma- Please see Appendix F.1 for a description.

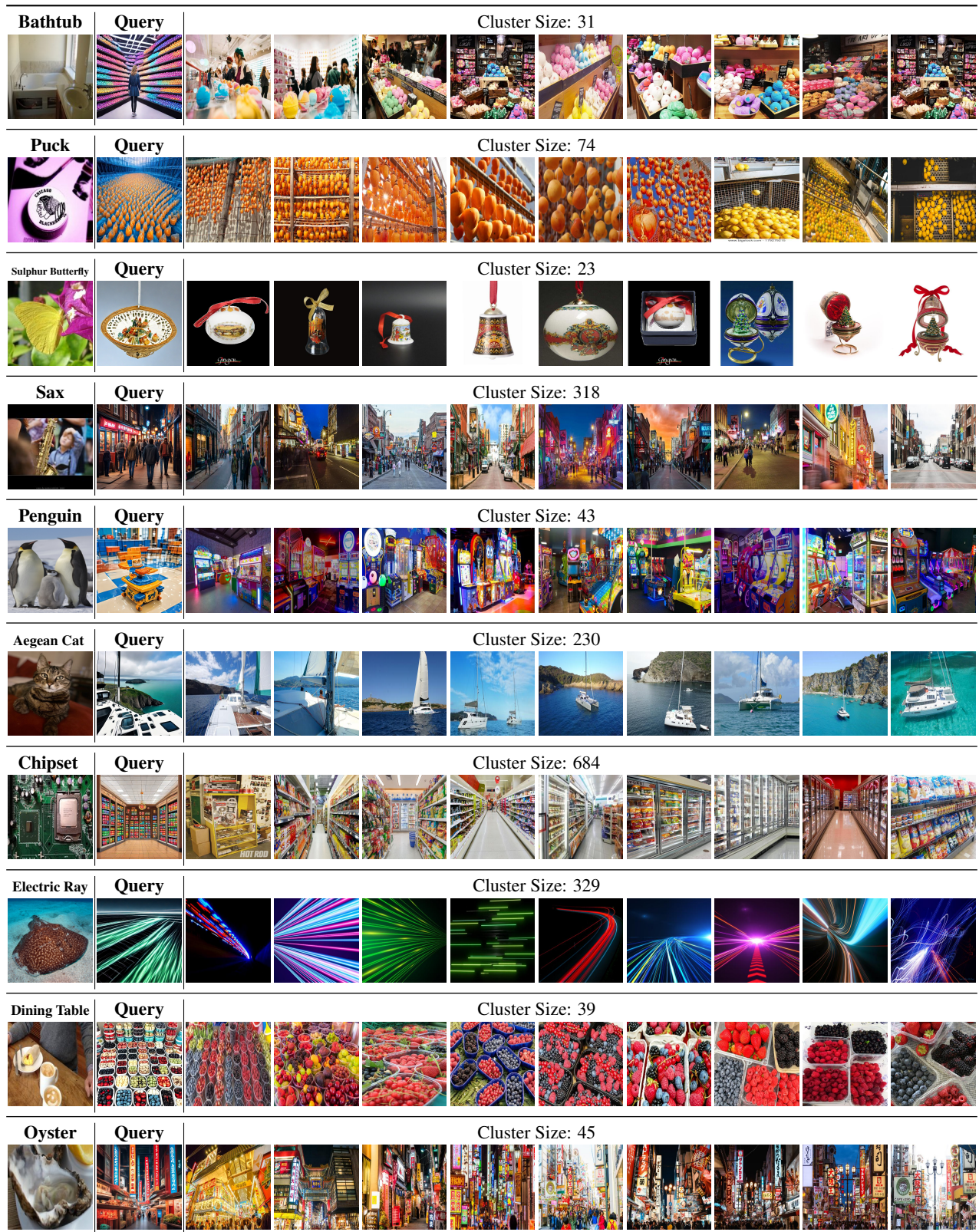


Figure 13. DASH-OPT PaliGemma- Please see Appendix F.1 for a description.

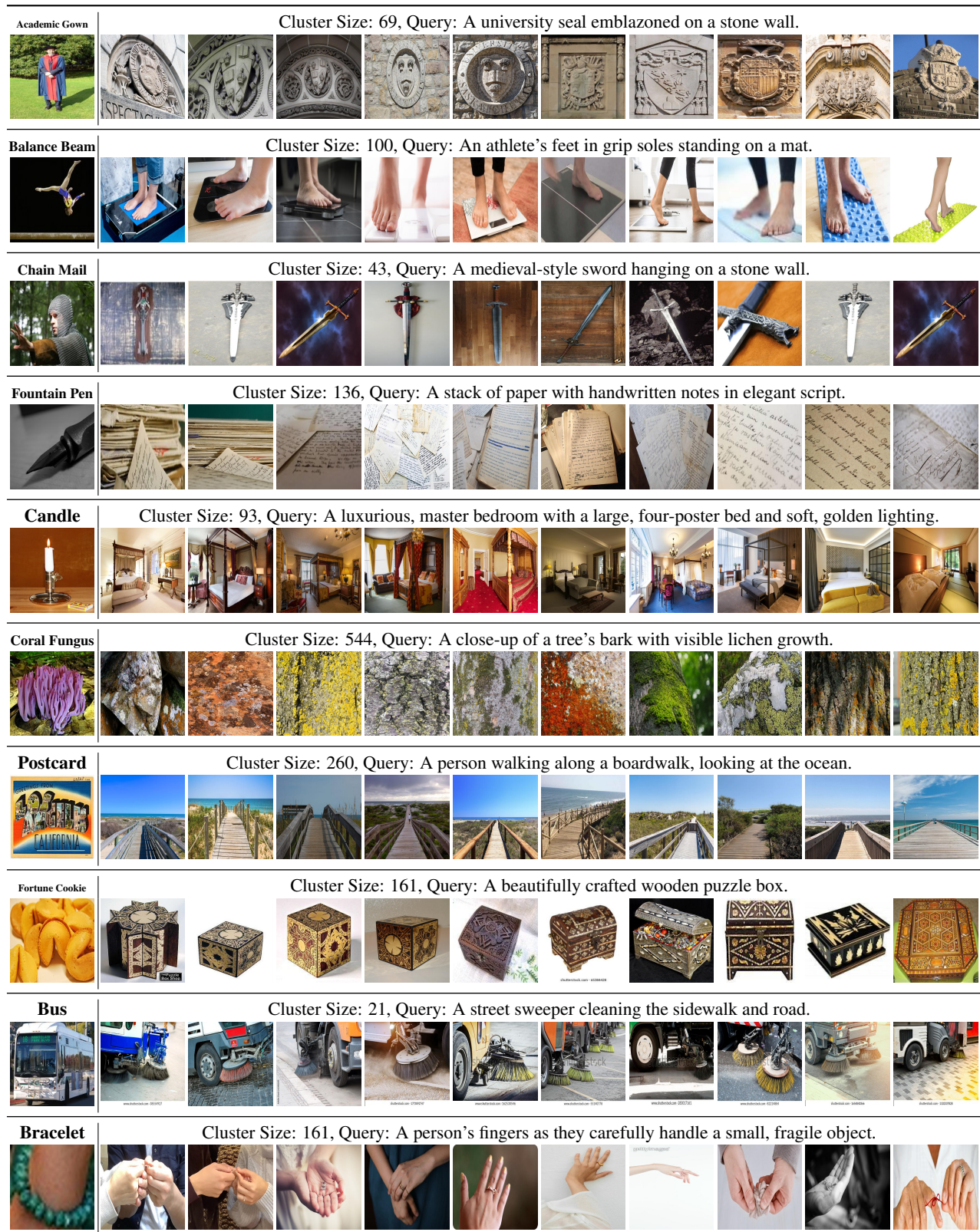


Figure 14. DASH-LLM- LLaVA-NeXT Vicuna- Please see Appendix F.1 for a description.

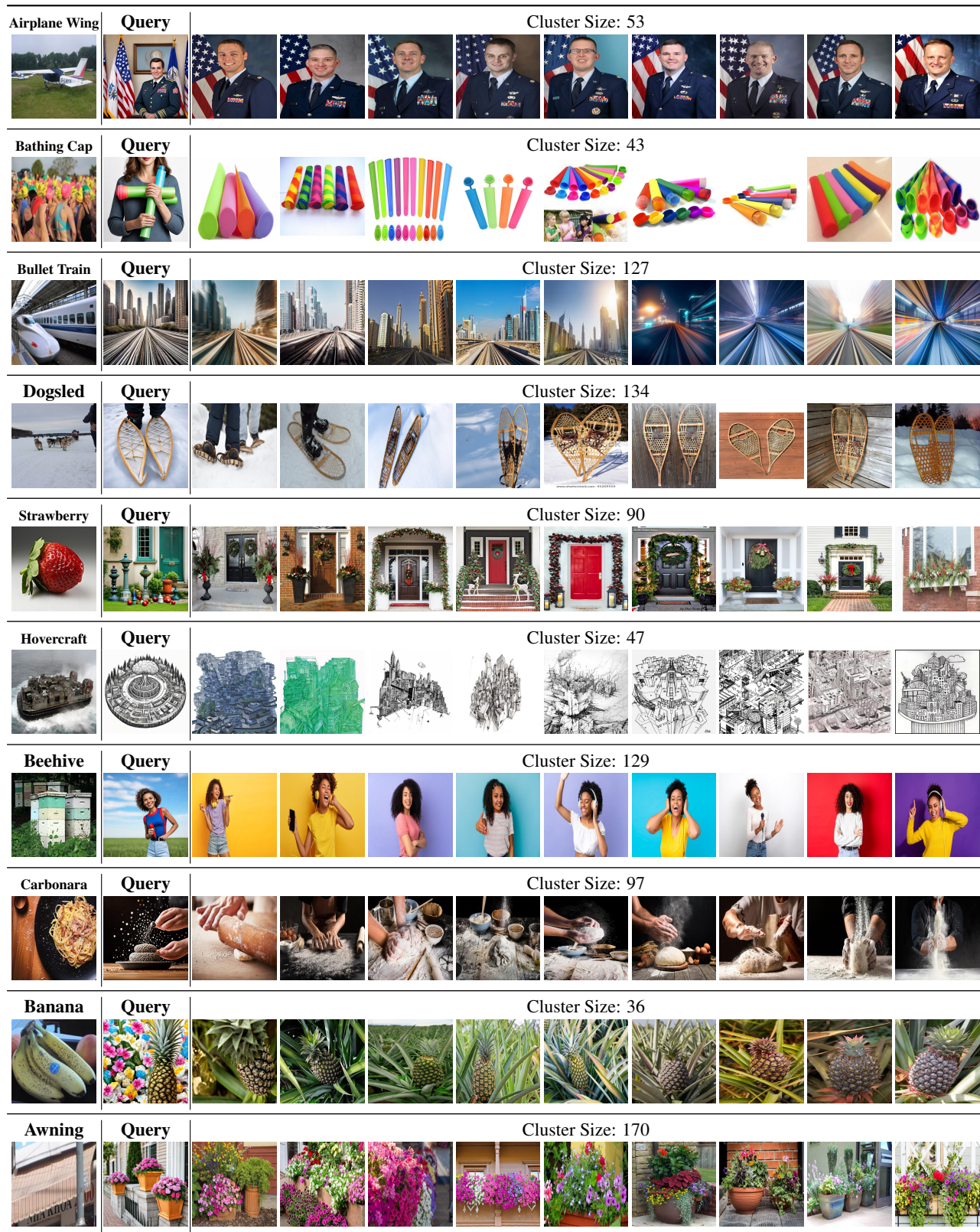


Figure 15. DASH-OPT LLaVA-NeXT Vicuna- Please see Appendix F.1 for a description.

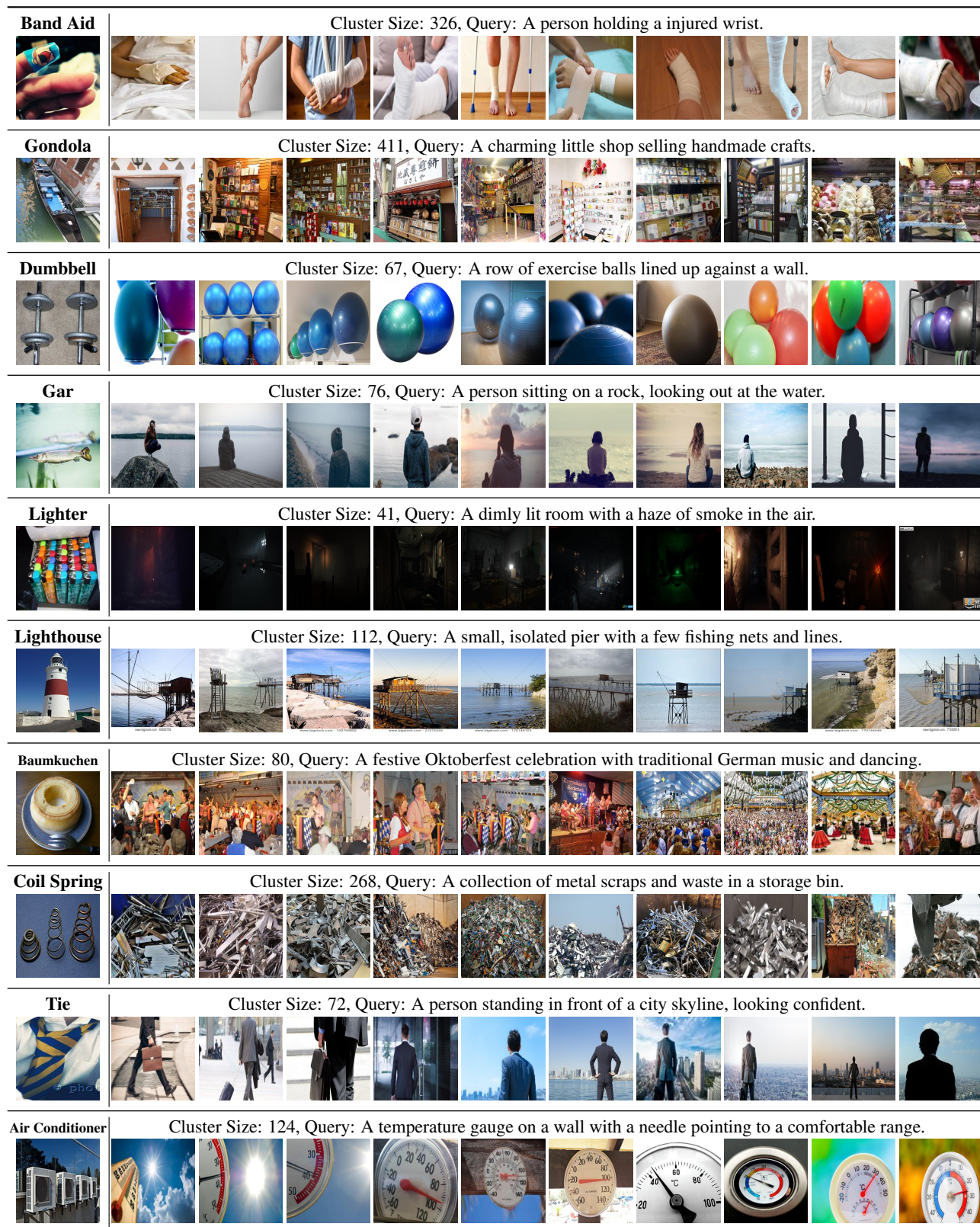


Figure 16. DASH-LLM LLaVA-NeXT Mistral - Please see Appendix F.1 for a description.

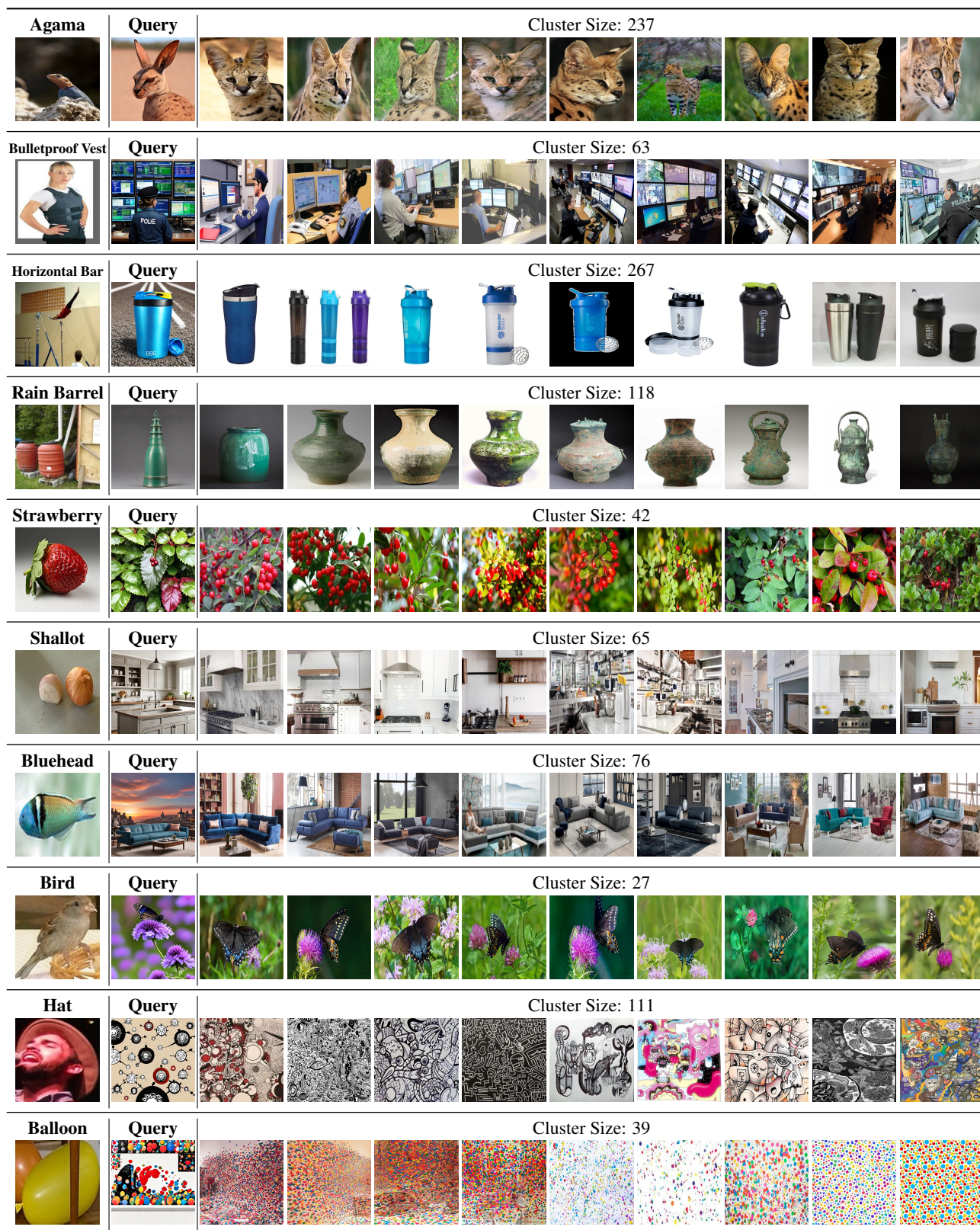


Figure 17. DASH-OPT LLaVA-NeXT Mistral- Please see Appendix F.1 for a description.

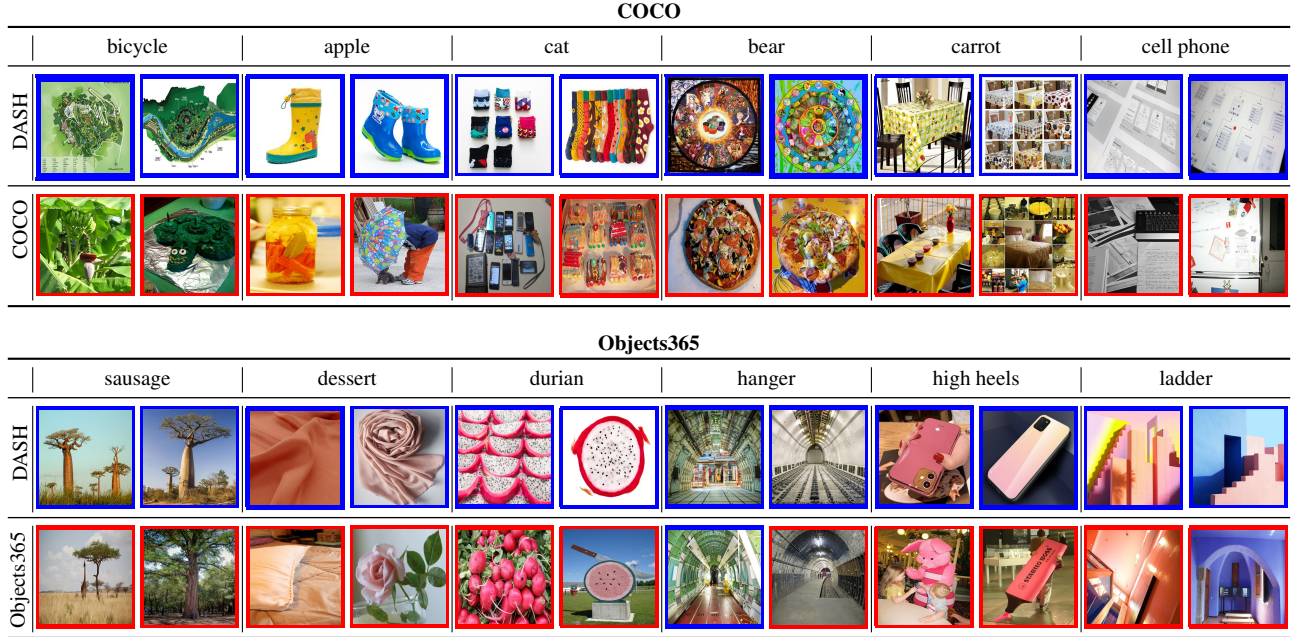


Figure 18. Demonstration of images that cause PaliGemma to detect the target class, identified using DASH-OPT, alongside their nearest neighbors in the reference datasets COCO and Objects365. For reference images, we use a blue border to mark images that elicit a "yes" response from the VLM and a red border for a "no" response. We show that neither the full COCO training set (80K samples) nor Objects365 (1.7M samples) contain the systematic errors uncovered by DASH, as all nearest neighbors are not detected by the VLM. This again highlights that our open-world search in ReLaion-5B is necessary to detect these hallucinations and would not possible even with such a reasonably large dataset such as Object365. With DASH we find that, PaliGemma incorrectly answers 'yes' for colorful 'wellington boots' as 'apple' and for 'Baobab trees' as 'sausage.'

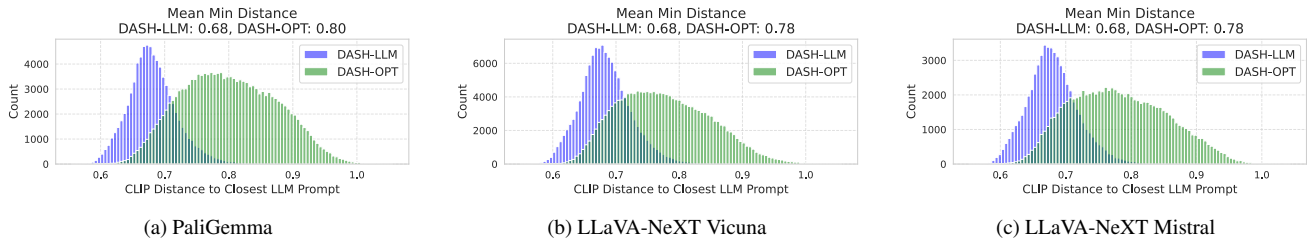


Figure 19. Extension of Fig. 4 for PaliGemma as well as LLaVA-NeXT Vicuna and Mistral. For all VLMs DASH-OPT finds hallucinations which are further away from the original text queries than DASH-LLM. This illustrates quantitatively the higher diversity of hallucinations found by DASH-OPT.

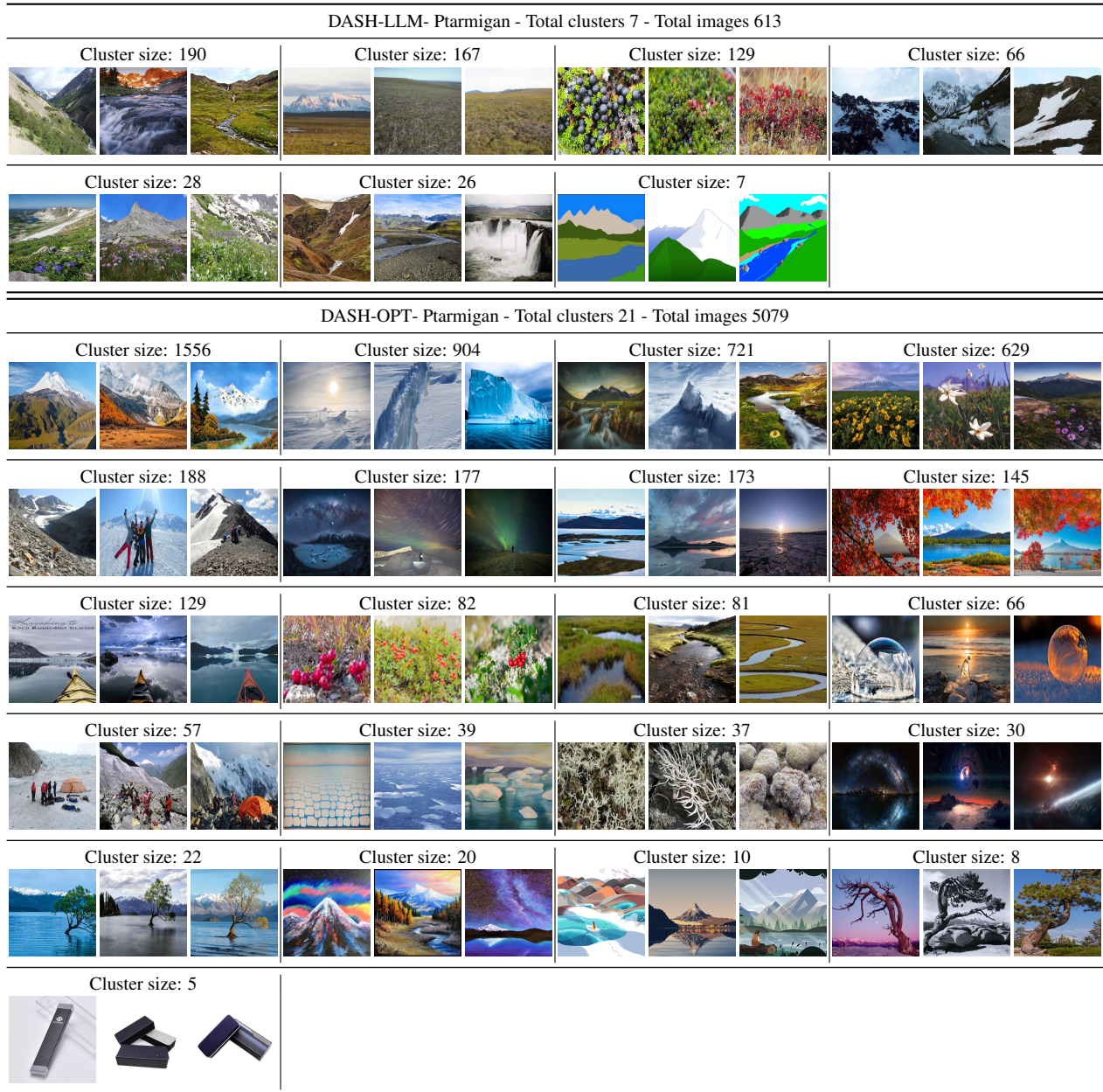


Figure 20. All clusters found for DASH-LLM and DASH-OPT for PaliGemma and the object “Ptarmigan”. While “Ptarmigan” refers to the bird species, the clusters include false positives such as images of mountain landscapes, alpine environments, and even abstract artistic representations or completely unrelated objects. This highlights how the VLM’s understanding conflates semantic and contextual cues with visual content, leading to hallucinations. In particular, we believe that these hallucinations could be caused by places containing the name “Ptarmigan,” such as multiple locations called “Ptarmigan Peak” in Colorado, Utah, and Alaska, or “Ptarmigan Ridge” and “Ptarmigan Traverse” in Washington. While we believe that a VLM should not respond that it sees a “Ptarmigan” even in an image of a place with a name containing the word “Ptarmigan,” we also checked several of these images to verify that these places are different mountainsides with completely unrelated names. This verifies that the VLM has learned a false representation of the word “Ptarmigan,” which includes many different mountainsides or peaks. Our DASH-OPT method, leveraging optimized queries, discovers additional “unknown unknowns,” such as rare or abstract scenes where a ptarmigan is highly unlikely (e.g., auroras, surreal artwork, and highly stylized objects). By creating queries for the specific target VLM, DASH-OPT uncovers vulnerabilities that are less intuitive or expected, revealing the VLM’s susceptibility to type II hallucinations.

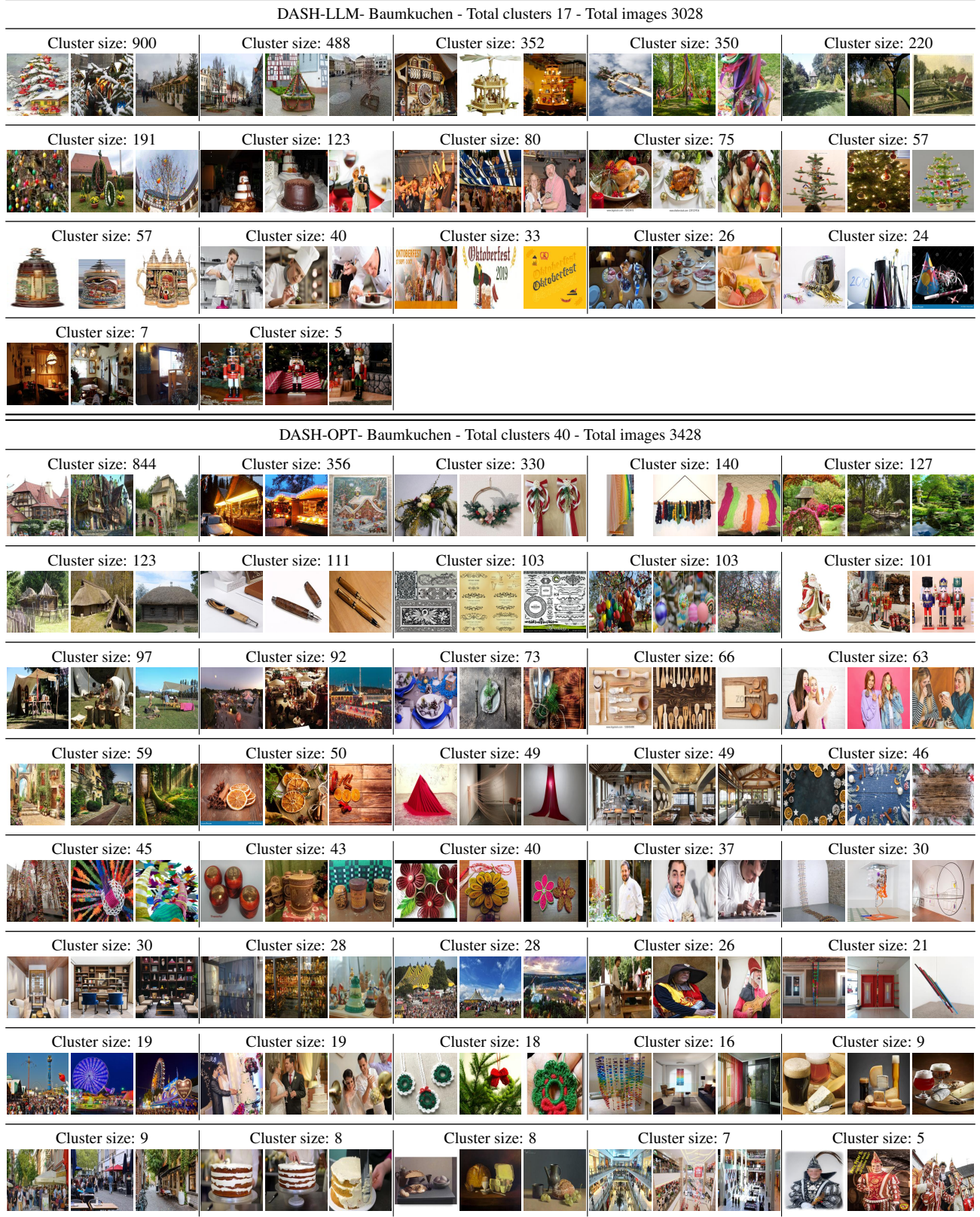


Figure 21. All clusters found for DASH-LLM and DASH-OPT for LLaVA-NeXT Mistral and the object "Baumkuchen". Please refer to Appendix F.2 for a description.

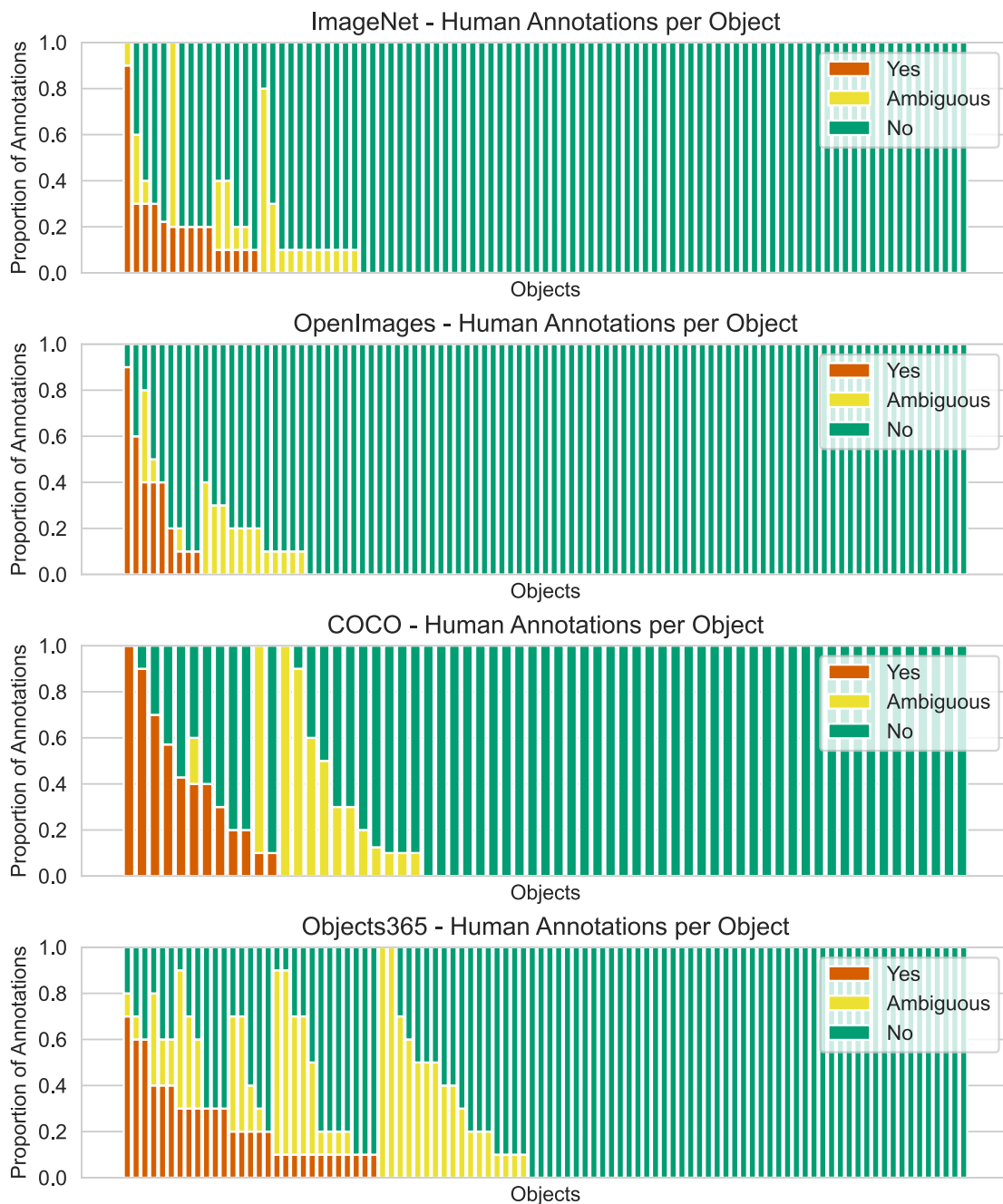


Figure 22. "yes", "no" and "ambiguous" rates in our human evaluation for the 4 datasets used for object labels in our evaluation. Each bar represents one object, for which we manually labeled 10 images for DASH-OPT on PaliGemma. We note that most objects do not contain any instances of the object and instead, most errors come from few object categories where the object detector itself has a systematic issue. We show qualitative examples in Fig. 23 and Fig. 24.

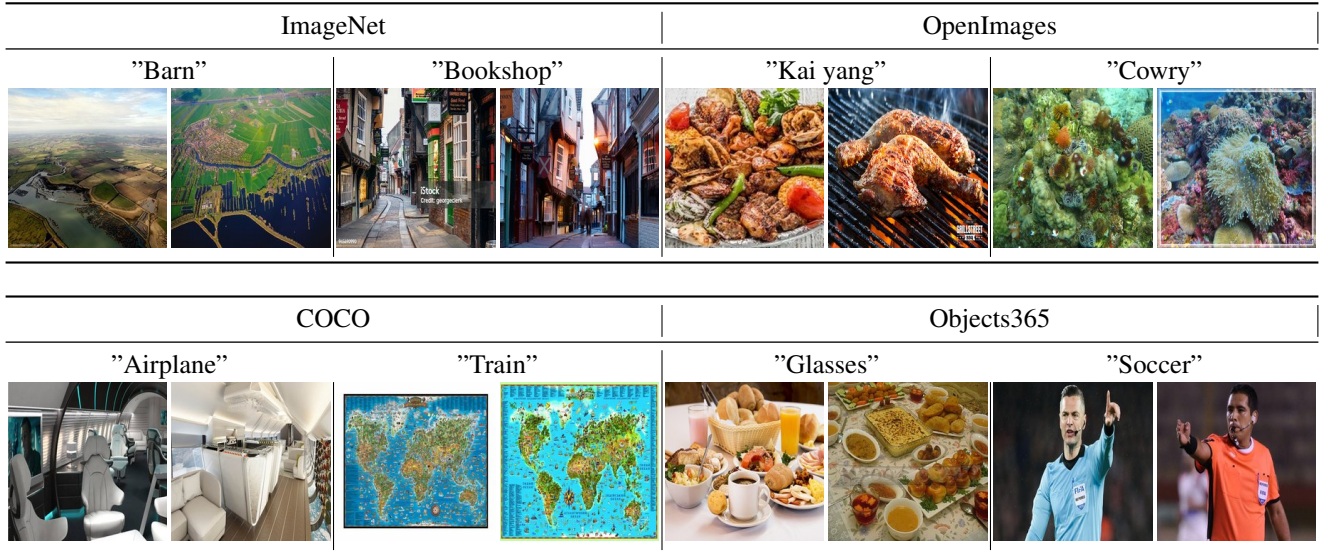


Figure 23. Examples of images labeled as "ambiguous" in our human evaluation are shown. For "Barn" and "Bookshop," the limited image resolution makes it difficult to identify the objects in the image; for instance, distinguishing a house from a barn in an aerial view is nearly impossible. For "Kai yang," a Thai dish with chicken, while the depicted dishes might contain chicken, it is challenging to determine whether they are specifically "Kai yang." Notably, a reverse image search labels the first image as "kebab." For "cowry," small sea snails, even if human labelers could not identify any, it is difficult to guarantee their absence in the image. For "airplane," the interiors shown could represent futuristic airplane or train designs. For "train," the image resolution is too low to infer the presence of specific objects. For the two objects from Objects365, the ambiguity mainly arises from the object labels themselves. For example, "glasses" in the dataset refers to eyewear, but the images often contain multiple glass objects. Similarly, "soccer" refers exclusively to a soccer ball in Objects365, whereas the sport itself is not a well-defined object. This creates ambiguity about whether we should label referees as "yes" or "no."

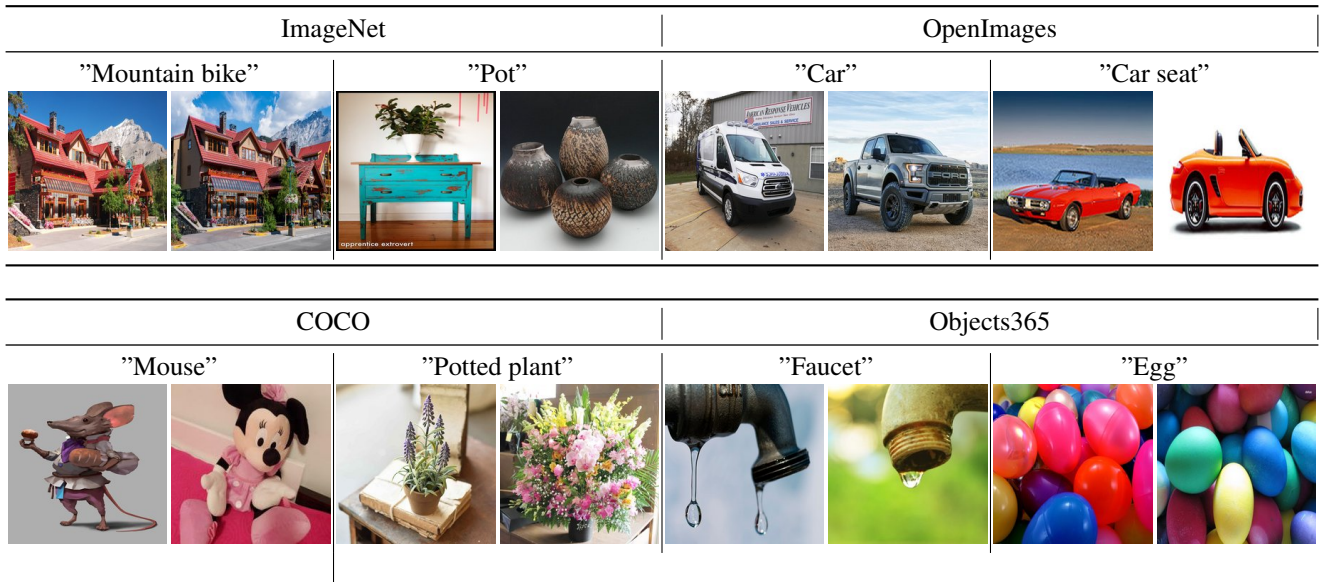
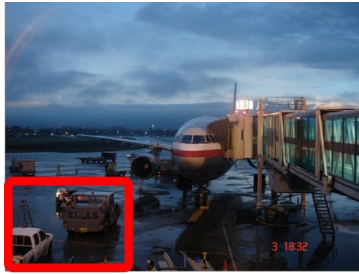


Figure 24. We highlight several failure cases of the object detector identified during our human evaluation. All images presented here have a confidence score below the threshold of 0.1 and are therefore not rejected by our automated pipeline. For "mountain bike," the primary issue arises from the objects being very small and difficult to spot. In the case of other objects, such as "Pot" or "Faucet," the objects are clearly visible, but the object detector fails to recognize these instances. For "Car," the detector does not seem to classify trucks or vans as cars. For "Mouse" and "Egg," the detector struggles with distribution shifts, failing to recognize comic or plush mice and colored eggs, respectively.



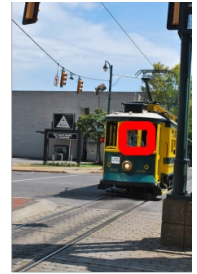
Car



Dining Table



Car



Person

Figure 25. **COCO annotations errors in POPE (ground truth “no”):** We show four examples where the POPE ground truth label for the question “Is there a *object* in the image?” is “no” although the object is present in the image. We mark the location of the object with a red bounding box.

I. Transfer

I.1. VLM Models

For all models except Prismatic, we use the Transformers library [56] with the official checkpoints. For Prismatic [17], we use the official implementation. Model details, including links to the specific models files used can be found in Tab. 5. Note that Qwen2-VL is not based on SigLIP as stated in Tab. 2 but instead uses a custom ViT with 675m parameters. We will correct this in the final version.

I.2. True positive rate

For each object from ImageNet, COCO, and Objects365, we collect 100 images of the corresponding class from the official validation set. On these images, the average TPR is computed by counting the frequency of the correct response “yes”.

I.3. Qwen2-VL vs Llama 3.2-VL

In the main paper, we have already shown some examples from DASH-B where, by design, both Qwen2-VL and Llama 3.2-VL hallucinate. While both these models are quite robust to hallucinations, we also want to understand where they differ. To do this, we show several examples in Fig. 26 where only one model hallucinates. This demonstrates that even the best available open-weight models are still vulnerable to hallucinations but also differ substantially in terms of vulnerabilities, likely due to larger differences in architecture, vision encoder, LLM, and training data.

J. DASH-B

As described in H, most models only produce a small number of false positives on POPE which also contain a large amount of label errors. Therefore, we propose a new benchmark DASH-B based on our retrieval results to enable a more reliable and rigorous evaluation of object hallucinations. Tab. J contains results for DASH-B and POPE for a range of VLMs.

J.1. Image Selection

We select the images for the benchmark using the following steps:

- We merge the images found by DASH-LLM and DASH-OPT over all three source models.
- These images are filtered by requiring a successful transfer to both Qwen2-72B and Llama 3.2-VL-11B, the best performing models in 2, in order to exclude errors which are specific to biases of the three source models.
- We select 70 objects and two human labelers verify that the selected images do not contain the corresponding object.
- The number of images is limited to at least 3 and at most 50.

- For each object, the same amount of positive samples, i.e. images that contain the object, are added. These images are retrieved using the Flickr API [10] and annotated by a human labeler to ensure that the object is clearly contained.

J.2. Metrics

The performance measure on DASH-B is the accuracy over all negative and positive samples. In Tab. J, we also report the true negative rate (TNR) and true positive rate (TPR) individually. A downside of measuring accuracy is that a trivial model that always replies “yes” (or always “no”) achieves an accuracy of 50%. This behaviour can be avoided by considering the harmonic mean of TNR and TPR instead which results in a value of 0 for the trivial case. We also report this metric (HM) in Tab. J but observe no significant effect on the results (apart from LLaVA-NeXT-Vicuna). Note that the results for the three source models are biased as they were used in the creation of the benchmark. Similarly, Qwen2-72B and Llama-3.2-11B are not reported as they produce a TNR of 1.0 by design.

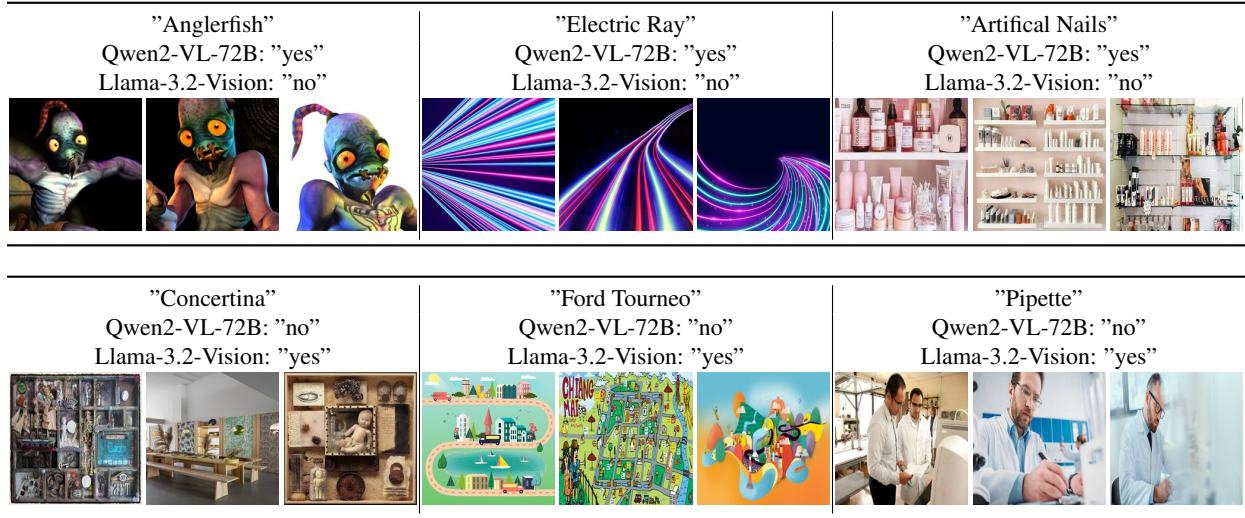


Figure 26. We demonstrate several images where Qwen2-VL-72B and Llama-3.2-Vision disagree. Note that all images do not contain the object and are thus hallucinations by the model responding with "yes". This further demonstrates that even the best available open-weight models are not robust to hallucinations.

VLM Model	LLM	Vision Encoder	Checkpoint
PaliGemma-3B [3]	Gemma-2B [49]	SigLIP-So400m 224px [60]	paligemma-3b-mix-224
LLaVA-NeXT-Mistral-7B [27, 28]	Vicuna-7B [36]	CLIP ViT-L 224px [38]	llava-v1.6-vicuna-7b-hf
LLaVA-NeXT-Vicuna-7B [27, 28]	Mistral-7B [16]	CLIP ViT-L 224px [38]	llava-v1.6-mistral-7b-hf
LLaVA-NeXT-Llama-8B [22, 27]	Llama-3.0-8B [9]	CLIP ViT-L 224px [38]	llama3-llava-next-8b-hf
Prismatic CLIP [17]	Vicuna-7B [36]	CLIP ViT-L 224px [38]	prismatic-vlms/clip-224px+7b
Prismatic SigLIP [17]	Vicuna-7B [36]	SigLIP-So400m 224px [60]	prismatic-vlms/siglip-224px+7b
Prismatic DinoV2 [17]	Vicuna-7B [36]	DINOv2 ViT-L 224px [35]	prismatic-vlms/dinov2-224px+7b
Qwen2-VL-7B-Instruct	Qwen2-7B [58]	Custom ViT 675m	Qwen2-VL-7B-Instruct
Qwen2-VL-72B-Instruct	Qwen2-72B [58]	Custom ViT 675m	Qwen2-VL-72B-Instruct
Llama-3.2-11B-Vision-Instruct	Llama-3.1-8B [9]	Custom ViT	Llama-3.2-11B-Vision-Instruct

Table 5. VLMs used for transfer experiments

Benchmark Metric	POPE	DASH-B			
	Acc.	Acc.	TNR	TPR	HM
PaliGemma-3B [3]	87.2%	62.0%	26.4%	97.7%	41.6%
LN Vicuna [28, 36]	87.6%	53.7%	10.4%	96.9%	18.7%
LN Mistral [16, 28]	88.0%	61.7%	30.1%	93.4%	45.5%
LN Llama [9, 28]	88.0%	65.2%	37.0%	93.4%	53.0%
Llava-OneVision [23]	88.7%	75.1%	60.2%	90.1%	72.2%
PaliGemma-2-3B [48]	88.8%	68.9%	40.9%	96.8%	57.5%
PaliGemma-2-10B [48]	87.7%	69.8%	48.0%	91.6%	63.0%
Ovis2-1B [30]	88.9%	64.6%	35.1%	94.0%	51.1%
Ovis2-2B [30]	89.4%	61.7%	27.3%	96.1%	42.5%
Ovis2-4B [30]	90.3%	64.8%	31.0%	98.6%	47.2%
Ovis2-8B [30]	94.9%	71.4%	44.8%	98.0%	61.5%
InternVL2.5-8B [4] *	90.6%	71.7%	47.2%	96.2%	63.3%
InternVL2.5-26B [4] *	90.6%	77.5%	57.3%	97.8%	72.2%
InternVL2.5-38B [4] *	90.7%	76.2%	54.8%	97.6%	70.2%
InternVL2.5-78B [4] *	90.8%	74.1%	50.3%	97.8%	66.5%
InternVL2.5-8B-MPO [4] †	89.1%	69.4%	42.3%	96.4%	58.8%
InternVL2.5-26B-MPO [4] †	90.7%	76.1%	54.8%	97.4%	70.1%
GPT-4o-mini*	84.2%	86.3%	77.0%	95.7%	85.3%

*: POPE result from [4], †: POPE result from [8]

Table 6. **DASH-B:** We report accuracy (for POPE and DASH-B) as well as the true negative rate (TNR), true positive positive rate (TPR), and the harmonic mean of TNR and TPR (HM). While the accuracy reflects the detection-hallucination trade-off, the individual values of TNR and TPR can give further insides into the vulnerability to FP-hallucinations. Note that PaliGemma-3B, LN Vicuna, and LN Mistral were used in the creation of the benchmark.

K. Fine-tuning on DASH

Can we utilize the images retrieved by DASH to mitigate the vulnerability to systematic hallucinations? To test this hypothesis, we perform a small scale experiment by fine-tuning PaliGemma-3B with LoRA[14] on our retrieval results. Used hyperparameters are provided in Tab. 7.

K.1. Data

DASH retrieves images, where the object is not present in the image. Therefore, the ground truth answer to the question “Can you see an *object* in this image?” is always “no”. We additionally retrieve images containing the object and add them to the training data to preserve the model’s ability to recognize the object. For each object, we add 200 negative samples, i.e. images where “no” is the correct reply, and 400 positive samples, i.e. “yes” is the correct reply, at random to the training set.

Negative samples: For the negative samples, i.e. images where the ground truth answer is “no”, we use all images resulting from DASH-LLM and DASH-OPT (both for PaliGemma). We split these images into two disjoint subsets:

- *Validation:* For each object, one of the found clusters is selected and all corresponding images are placed in the validation set.
- *Train:* All remaining images are used to sample images for the fine-tuning dataset. We further filter these images to ensure that they do not contain the object by requiring that Llama 3.2-VL and Qwen2-VL answer with “no”.

Positive samples: We generate a diverse set of prompts including the objects using Llama 3.2 and use them to retrieve images from ReLAION. The resulting images are filtered by the object detector (threshold > 0.1) and Llama 3.2 (response “yes”).

Optimizer	ADAM
β_1	0.9
β_2	0.999
Learning rate	1e-6
Number of epochs	5
Batchsize	32
LoRA rank	8

Table 7. Fine-tuning hyperparameters

K.2. Results

We report several metrics for PaliGemma-3B and our fine-tuned version (+ft) in Tab. 8, comparing their performance on different tasks:

- **Systematic hallucinations:** The accuracy, i.e. ratio of correctly replying with “no”, on the Validation set.

- **Hallucination benchmarks with similar tasks:** We report the Amber score and the accuracies on Amber Existence and R-Bench.
- **Effect on other tasks:** We evaluate two VQA benchmarks (TextVQA [46], VQAv2 [12]) and two captioning benchmarks (COCO[25], Flickr30k [59]) and report accuracies and CiDER scores, respectively.
- **Performance on positive samples:** The TPR-ICO for the objects from ImageNet, COCO, and Objects365 are evaluated as described in Appendix I.2.

The fine-tuned version (+ft_{pre}) significantly improves over PaliGemma-3B on unseen clusters (Validation, +77.7%). It also shows slightly better results on related hallucination benchmark with increases in the Amber score (+0.2), as well as higher accuracies on Amber Existence (+3%) and R-Bench (+1.1%). The performance decreases slightly for more general VQA tasks (−1.4% and −0.9%) and captioning tasks (−1.3 and +0.1%). The reduction of the TPR-ICO (−5.1%) is due to a significant drop of the TPR on Objects365 (−12.1%). A possible reason for this is a mismatch between the image distributions of the retrieved positive samples, where the object is prominently visible in the image, and Objects365, where objects often occur only in small bounding boxes inside the image. Evidence for this is shown in the last column (+ft) of Tab. 8: We repeated the fine-tuning on a different dataset where we replaced all retrieved positive samples for objects from Objects365 with images from the original Objects365 training set. In this setting with more positive than negative samples, the fine-tuned model even improves TPR on all three datasets but also improves less on the hallucination tasks. This experiment indicates that the images retrieved by DASH can also be used to mitigate the problem of systematic hallucination by including them into a fine-tuning routine.

L. Reverse Task

We apply the DASH-LLM pipeline to the reverse task, where the VLM outputs “no” despite the object being visible in the image. We adjust the LLM prompt accordingly (see 28), reverse the object detector threshold, and use a larger value. Figure 29 presents example clusters. While this experiment serves as a proof of concept, we observe that the object detector performs worse in this direction and should be replaced for larger-scale experiments. Overall, the benefits of DASH are more pronounced in the setting discussed in the main paper, as the number of images containing a given object is much smaller than the number of images that do *not* contain the object.

Dataset	Metric	PaliG	+ft _{pre}	+ft
Validation	Acc.	0.0%	77.7%	57.6%
Amber	Score	93.5	93.7	94.0
Amber Ex.	Acc.	93.2%	96.2%	95.4%
R-Bench	Acc.	79.9%	81.0%	80.2%
TextVQA	Acc.	57.6%	56.2%	56.5%
VQAv2	Acc.	83.1%	82.2%	82.4%
COCO	CiDER	124.5	123.2	121.3
Flickr30k	CiDER	77.4	77.5	77.1
ImageNet	TPR	90.0%	90.0%	93.4%
COCO	TPR	84.0%	80.4%	88.8%
Objects365	TPR	69.0%	56.6%	73.0%
TPR-ICO	TPR	81.1%	76.0%	85.1%
DASH-B	Acc.	56.4%	-	68.0%
DASH-B	TNR	26.4%	-	45.9%
DASH-B	TPR	86.4%	-	90.0%

Table 8. Accuracies on our Validation set, Amber Existence, and R-Bench and TPR on positive samples from the validation sets of ImageNet, COCO and OpenImages. Fine-tuning on DASH results (+ft_{pre}) can improve robustness against hallucinations significantly, even on clusters not seen during training. It also improves on related hallucination benchmarks while the performance on more general VQA and captioning tasks becomes slightly worse. The reduction in TPR-ICO is caused by the retrieved positive samples for Objects365. After replacing these with images (+ft) from the original training set of Objects365, the fine-tuning even improves average TPR-ICO.

```

1 You are a creative prompt generator. Your task is to:
2
3 1. Accept an object name (provided by the user).
4 2. Generate 20 different image prompts in realistic everyday settings, filled with various common
   objects, where the specified object is present but not necessarily the focus of the scene-so it
   might be overlooked by an object detection system.
5
6 ---
7
8 ## Context & Objectives
9
10 1. Purpose:
11 - We want to depict the given object in real-life scenarios that include multiple other items
   typically found in the setting.
12 - The object should be there, but the scene should be busy or populated enough that the object isn't
   the sole focus.
13 - The style should be highly realistic, as if taken by a camera.
14
15 2. Guiding Techniques:
16 - Crowded Scenes: Combine the specified object with many other objects commonly found in the
   same environment (e.g., living rooms, offices, kitchens, garages).
17 - Non-Focal Positioning: Place the object off to the side or partially in the background, so it
   doesn't immediately draw attention.
18 - Realistic Keywords: To enhance the lifelike quality, you can use any of these keywords in your
   prompts:
19 - photo-real
20 - hyper-detailed
21 - 8k resolution
22 - cinematic lighting
23 - DSLR
24 - natural lighting
25 - raw photo
26 - high dynamic range
27 - real-world texture
28 - unposed
29
30 ---
31
32 ## Detailed Instructions
33
34 1. Input:
35 You will receive a single word or short phrase specifying the object (e.g., "chair," "cup," "clock,"
   "bag," etc.).
36
37 2. Output:
38 - Produce 20 unique prompts, each describing a realistic photograph in which the object is
   present among various other items typically found in that scenario.
39 - Use some of the realism keywords to convey a high-quality, real-world style.
40 - Ensure the object is not the main focus but simply part of a busier environment.
41
42 3. Format:
43 - Number each prompt from 1 to 20, using a colon (e.g., '1: Prompt text', '2: Prompt text', ...,
   '20: Prompt text').
44 - Each prompt should be concise but mention multiple items and the general setting.
45
46 ---

```

Figure 27. DASH-LLM prompt for generating the text queries for the reverse task (1/2)

```

1  ## Examples of Prompts
2
3
4  *(Using '<OBJECT_NAME>' as a placeholder - these are short samples, not fully detailed.)*
5
6  - **Living Room Scenario**
7    *"A photo-real image of a cozy living room with a sofa, coffee table, TV, potted plants, and a small
8      '<OBJECT_NAME>' tucked beside a stack of magazines."*
9
10 - **Office Setting**
11    *"A hyper-detailed view of an open-plan office featuring desks, laptops, file cabinets, a water
12      cooler, and a '<OBJECT_NAME>' placed casually near a window sill."*
13
14 - **Kitchen Scene**
15    *"A raw photo of a busy kitchen counter with plates, utensils, fruits, and a '<OBJECT_NAME>' resting
16      behind a jar of spices."*
17
18 Please use many different such scenarios instead of restricting yourself to the ones from these
19 examples.
20 Possible scenarios would be an office, a train station, a garden, a living room, a kitchen, a hallway,
21 outdoors, in the city, landscape.
22 Try to think of a scenario that matches the object and that allows you to add in different objects that
23 could occur with it.
24
25 Please try out different scenarios for each object in the different prompts.
26 Make sure to not repeat too similar prompts and rather create a sufficient variety of prompts.
27
28 These examples show:
29 - The '<OBJECT_NAME>' is included but not emphasized.
30 - The setting has multiple other common objects.
31
32 ---
33
34 ## Final Output Format
35
36 When the user provides the object name, respond with exactly **20 prompts**, numbered with colons, in
37 the form:
38
39 1: [Prompt text]
40 2: [Prompt text]
41 ...
42 20: [Prompt text]
43
44 Each prompt should describe a realistic scene filled with everyday objects, incorporating the given
45 object without making it the sole focus.

```

Figure 28. DASH-LLM follow-up prompt for generating the text queries for the reverse task (2/2)

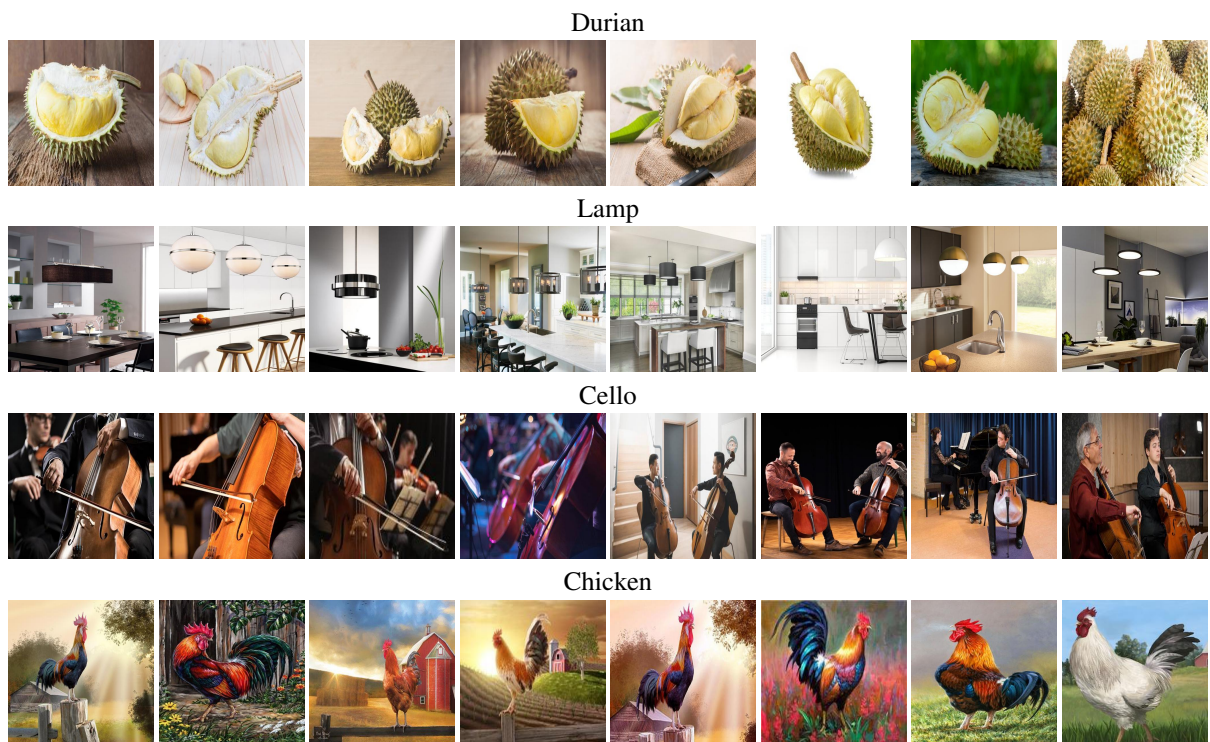


Figure 29. **Reverse Task** We show three clusters found by DASH for the reverse task using LLaVA-NeXT Vicuna: The VLM response to "Can you see a OBJ in this image?" is "no" although the object is clearly visible in the image.

M. Transfer across prompts

During our experiments we use the prompt “Can a *object* be seen in the image?”. In Tab. 9, we evaluate a range of 10 different prompts for the three source models on their corresponding DASH-LLM and DASH-OPT subsets.

	DASH-LLM			DASH-OPT		
Prompt Transfer Rate	PaliGemma	LN Vicuna	LN Mistral	PaliGemma	LN Vicuna	LN Mistral
Can a OBJ be seen in the image?	0.861	0.923	0.824	0.878	0.872	0.832
Does this image have a OBJ?	0.890	0.891	0.806	0.859	0.863	0.781
Does the image show a OBJ?	0.860	0.737	0.733	0.832	0.712	0.733
Does this image contain a OBJ?	0.857	0.790	0.858	0.801	0.739	0.843
Does this picture include a OBJ?	0.824	0.782	0.886	0.795	0.735	0.865
Is a OBJ depicted in this image?	0.851	0.913	0.748	0.805	0.901	0.755
Is there a OBJ present in the image?	0.787	0.807	0.860	0.758	0.735	0.870
Is there a OBJ in this image?	0.717	0.818	0.827	0.643	0.771	0.828
Is a OBJ shown in the image?	0.670	0.880	0.792	0.593	0.835	0.804
Is OBJ visible in the image?	0.467	0.833	0.655	0.385	0.782	0.655
Is OBJ in the image?	0.348	0.906	0.702	0.278	0.873	0.690
Average	0.739	0.844	0.790	0.693	0.802	0.787
Standard Deviation	0.179	0.062	0.072	0.200	0.069	0.071

Table 9. **Transfer across prompts:** While transfer rates for LN Vicuna and LN Mistral are stable, PaliGemma was pretrained on this task using the prompt "Is OBJ in the image?" and shows lower transfer rates on similar prompts. However, this improved robustness against systematic hallucinations does not generalize to less similar prompts.