Multimodal Large Language Models as Image Classifiers

Supplementary Material

A. Related Work

MLLM evaluation. MLLMs benchmarks [13, 17, 22, 23] have largely converged on multiple-choice question answering as the dominant evaluation format, favored for its simplicity and unambiguous scoring. However, this format makes direct comparison with traditional vision models (VLMs, supervised classifiers) difficult, as it measures different capabilities. Some benchmarks [17, 22] design more challenging distractors, while others [14] argue that selecting from a list is fundamentally easier than generating an answer, promoting self-retrieval methods that test a model's ability to describe and distinguish images without provided options. Our work fills this gap by evaluating MLLMs on standard image classification under conditions comparable to traditional vision models.

MLLMs in Image Classification. Zhang et al. [45] show that generative MLLMs perform far below CLIP on ImageNet-1k, even in a Closed-World task, though limited to 100 classes due to token length constraints of older models. To address OOP predictions, they introduce Probabilistic Inference, which constrains generation to only tokens from the provided class list. While effective, this is computationally intensive: for 1,000 classes, inference is 1,000 times slower than direct generation. This contrasts with our CW+ approach, which resolves OOP via fast post-processing. The authors also evaluate an OW task, using string matching (*i.e.* text inclusion) to map free-text predictions to class names. This mapping strategy is inefficient, which explains why they find OW to always perform worse than CW — a finding we do not replicate with our embedding-based mapping.

Liu et al. [21] report that newer MLLMs approach or surpass CLIP-like systems, though typically under MC conditions. They explore increasing the number of answer options up to 26 — the number of letters in the alphabet — and find that accuracy decreases as options grow. To increase difficulty, they construct harder distractors using the top-25 most semantically similar class names identified with BERT [10], observing only a slight accuracy drop of 2.3 percentage points for Qwen2-VL on ImageNet-1k. Notably, their experiments are run once without confidence intervals. Our confidence-interval-backed experiments show a considerably larger accuracy drop under their sampling strategy, and a further increase with our harder distractor selection strategy. Similarly to [45], the authors do not evaluate on the full class list due to OOP, a limitation we address with CW+.

Conti et al. [7] evaluate MLLMs in an Open-World task across several metrics, including cosine similarity between model predictions and ground-truth labels in the Sentence-BERT [31] embedding space, which is an inspiration for our embedding-space approach, but we utilize newer models. We additionally include experiments with their encoder in Tab. 17. However, they do not report results for ImageNet-1k and focus solely on MLLMs and the OW task, whereas we also observe performance on other tasks, and compare it against VLMs and supervised models.

B. Case study: ChatGPT vs. humans

We conducted a second annotation pass by randomly assigning all challenging S− (1894) and M− (1078) images to the annotators. For each image, they received the following in random, anonymized order: 1. GPT-4o prediction, 2. ReGT, 3. ImGT, and 4. SigLIP 2 prediction. GPT-4o was prompted to return all ImageNet-1k labels present in an image as class IDs rather than full class names, reducing token usage and associated costs.

ChatGPT accuracy on S− and M− images was initially 34.37 % and 34.32 %, respectively, with respect to ReGT. After the second pass, accuracy increased to 55.20 % and 55.19 %. An analysis of the results from this verification pass is presented in Fig. 7. For 52.6 % of S− and 49.5 % of M− images (■ + ■ sections in Fig. 7), annotators either confirmed the ChatGPT prediction as the only correct label or combined it with ReGT label.

There remains a set of 30.9 % S− and 47.5 % M− mispredicted images (■ sections in Fig. 7) where at least some of ReGT labels were preserved and no ChatGPT predictions were added. Examples of these reannotations are presented in Fig. 5. We conclude: 1. Error corrections made by trained annotators are not entirely reliable, being completely incorrect (■ + ■) in 50.6 % of S− and 8.7 % of M− images; and 2. GPT-4o can serve as a valuable assistant for flagging such errors for additional verification.

C. Evalutation Setup - Details

All MLLMs evaluated in our experiments, along with their corresponding vision backbones and language encoders, are summarized in Tab. 7.

C.1. Prompt overview

The exact prompts used for InternVL3.5, LLaVA-OV, Qwen3-VL, and GPT-4o are provided in Tab. 19 for the CW and CW+ tasks, and in Tab. 20 for the OW task. In the MC setup, all models share the same prompt, as shown in Tab. 21, whereas the OW and CW prompts differ. A key

Model	Vision Backbone	Lang. Encoder	Training Strategy
PaliGemma2-mix-28B/448	SigLIP so400M	Gemma2 27B	Joint multimodal pretraining + finetuning
LLaVA-OneVision-72B-Chat	SigLIP so400M	Qwen2 72B	Projector-based alignment + supervised finetuning
InternVL3.5-38B	InternViT 6B	Qwen3 32B	Progressive multimodal scaling + instruction tuning
Qwen3-VL-235B-A22B-Inst	SigLIP 2 so400M	Qwen3 MoE	Large-scale multimodal pretraining + instruction tuning
GPT-4o-2024-08-06	(undisclosed)	(undisclosed)	Proprietary multimodal pretraining + post-training alignment (RLHF)

Table 7. Vision backbones and language models for all five evaluated MLLMs. GPT-4o-2024-08-06 is a closed-source model; architectural details are not publicly disclosed.

distinction is that GPT-4o prompts generally omit detailed instructions on output formatting, since the desired output structure is enforced directly via the API. For PaliGemma 2, the standard `describe en` prompt was employed in the OW setup. The CW task could not be performed due to the input prompt exceeding the token length limit.

C.2. Equal classes

We list all class pairs treated as equal for evaluation in the main paper, along with image examples and brief explanations, in Fig. 10 and Fig. 11.

C.3. Weasel family case study

The reanotated dataset, as introduced in Sec. 2.1, only contains 625 out of the original 1000 classes, where the majority of wildlife, with the prominent exception of about 120 dog breeds, is excluded.

This is because wildlife is notoriously hard to annotate for non-experts, even if the annotators are trained. To account for this limitation, we also introduce a case study on four classes from the weasel family: weasel, mink, polecat and black-footed ferret, whose reannotation conducted by an expert was introduced in [16]. For a brief overview of the classes and image examples, see Fig. 8.

The weasel problem. Here we provide a brief overview of the classes and the main issues. For more details, we refer the reader to the original publication.

Kisel et al. [16] revisited four closely related mustelid classes in ImageNet and found severe problems arising from ambiguous synsets, inconsistent taxonomy, and extensive mislabeling. Their expert reannotation shows that these fine-grained wildlife categories cannot reliably serve as ground truth. The main issues are:

- **weasel**: Synset corresponds to a broad colloquial category rather than a specific species; many images depict other mustelids (e.g., mink, ferrets), leading to more than half of the images being incorrect. However, the dominant species is the least weasel and highly similar species. In American English, the term weasel often refers to the weasel family as a whole, encompassing all four classes.
- **mink**: Although somewhat cleaner, many images still mix American and European mink and include other small mustelids due to visual similarity and poor source metadata.
- **polecat (Mustela putorius)**: The term "polecat" is ambiguous across English varieties and is also used for skunks; images contain a heterogeneous mix of species, and only about one-third match the intended European polecat. The European polecat is the ancestor of the domestic ferret, which makes them hard to distinguish, and they can also interbreed.
- **black-footed ferret (Mustela nigripes)**: Synset conflates the endangered wild species with domestic ferrets; almost all images show domestic animals rather than M. nigripes, leaving the class effectively without valid examples. The class is renamed to 'domestic ferret'.

These issues stem from WordNet synset definitions that do not align with real-world photographic data, making the original ImageNet labels highly unreliable for fine-grained species evaluation.

Performance on reannotated data. Across model families, accuracy generally increases when evaluated on the reannotated labels, with substantially larger gains on the 159-image WeaselGT subset, see Tab. 8. The improvements are most pronounced for MLLMs and VLMs, indicating that many of their apparent errors under ImGT labels stem from annotation noise. In contrast, supervised models show smaller gains, suggesting stronger anchoring to the original label space. Tab. 9 complements this by reporting class-level recall under ImGT labels versus WeaselGT labels. Taken together, the tables show that careful reannotation is essential for reliably evaluating model performance on closely related wildlife categories.

C.4. Preliminary experiments

Research on image classification with MLLMs remains limited, with only three methodologies proposed in the literature, corresponding to OW, MC, and CW tasks. There is little guidance on how to properly configure MLLMs for such evaluations, and no prior studies have systematically examined the impact of different parameter settings. Therefore,

(a) weasel

(b) mink

(c) polecat

(d) black-footed ferret

Figure 8. Weasel family case study: Correctly labeled ImageNet-1k images for "weasel", "mink", "polecat", and "black-footed ferret". (a) **Weasel** often refers broadly to *mustelidae* family, but images mainly depict least weasels. (b) **Minks** are easier to recognize but still confused with similar species. (c) European **polecat** is the ancestor of domestic ferret and they are extremely hard to classify. Often, they are recognized by the environment, as opposed by the animals' features. (d) **Black-footed ferret** images are frequently domestic ferrets, leading to renaming. See [16] for details.

		ImGT		ReGT	WeaselGT		
Model	Task	A_{31250}	A_{159} 🪶	A_{31250}	A_{159} 🪶	$S{+}_{112}$	$S{-}_{47}$
ImGT	CW	100.00	100.00	91.20 -8.8	70.44 -29.6	100.00	00.00
LLaVA-OV	CW	42.56	25.79	52.00 +9.4	28.93 +3.1	35.71	12.77
	MC	82.17	45.91	90.01 +7.8	61.01 +15.1	59.82	63.83
Intern-VL3.5	CW	63.68	44.65	72.96 +9.3	61.01 +16.4	60.71	61.70
	MC	85.89	55.97	90.53 +4.6	75.47 +19.5	75.89	74.47
Qwen3-VL	CW	64.16	40.25	72.16 +8.0	50.94 +10.7	54.46	42.55
	MC	86.32	64.15	90.77 +4.5	83.65 +19.5	86.61	76.60
GPT-4o	CW	74.69	64.15	81.32 +6.6	84.28 +20.1	88.39	74.47
	MC	88.31	67.92	92.52 +4.2	91.19 +23.3	92.86	87.23
DINOv3 (dino.txt)	CW	79.36	69.18	85.12 +5.8	86.79 +17.6	91.96	74.47
SigLIP so400M	CW	82.88	66.04	87.20 +4.3	88.05 +22.0	91.07	80.85
SigLIP 2 so400M	CW	83.36	66.67	87.68 +4.3	88.68 +22.0	90.18	85.11
SigLIP 2 giant	CW	83.84	67.92	86.24 +2.4	91.82 +23.9	92.86	89.36
DINOv3 (k-NN)	CW	83.36	71.07	87.20 +3.8	79.87 +8.8	87.50	61.70
EfficientNetV2-XL	CW	85.12	71.07	86.88 +1.8	78.62 +7.6	87.50	57.45
EfficientNet-L2	CW	87.52	75.47	89.12 +1.6	80.50 +5.0	91.07	55.32
EVA-02	CW	90.24	73.58	91.68 +1.4	80.50 +6.9	90.18	57.45

Table 8. Accuracy on the full 31250-image ReGT set and on the WeaselGT 159-image subset 🪶. Increase or decrease indicates the accuracy change under ReGT vs. ImGT. For WeaselGT, deltas denote the accuracy difference between WeaselGT vs. ImGT on the same 159-image subset 🪶. Gains are consistently higher on WeaselGT, especially for VLMs and MLLMs—indicating greater sensitivity to fine-grained label corrections than supervised models.

we present an ablation study on the parameters influencing image classification using MLLMs.

Image batch size. We evaluate different batch sizes by sending batches of 1, 5, and 10 images per request to the LLaVA-OV, Qwen3-VL, and GPT-4o models. We do not perform this evaluation for InternVL3.5, as it only accepts a single image per request. The results in Tab. 10 show that Qwen3-VL and GPT-4o maintain nearly identical accuracy across all batch sizes, whereas LLaVA-OV experiences a dramatic drop in accuracy with larger batch sizes. Since batch size 10 provides stable results for both GPT-4o and Qwen3-VL and is more resource-efficient, we select it for all remaining experiments. In contrast, LLaVA-OV is evaluated using batch size 1, as it would be unfair to assess it under conditions known to reduce its performance.

Effect of image position within a batch on accuracy. The impact of image position within a batch is presented in Tab. 12. Qwen3-VL and GPT-4o consistently maintain high accuracy regardless of image order, while LLaVA-OV experiences a significant drop in performance for images appearing later in the batch, consistent with its sensitivity to larger batch sizes.

	weasel		mink		polecat		domestic ferret	
	ImGT	WeaselGT	ImGT	WeaselGT	ImGT	WeaselGT	ImGT	WeaselGT
ImGT	100.00	84.62 -15.4	100.00	85.00 -15.0	100.00	57.14 -42.9	100.00	61.54 -38.5
LLaVA-OV	40.00	76.92 +36.9	60.00	65.00 +5.0	0.00	0.00 +0.0	0.00	0.00 +0.0
Intern-VL3.5	30.00	65.38 +35.4	32.00	37.50 +5.5	0.00	0.00 +0.0	92.00	100.00 +8.0
Qwen3-VL	36.00	76.92 +40.9	72.00	80.00 +8.0	0.00	3.57 +3.6	42.00	43.08 +1.1
GPT-4o	46.00	100.00 +54.0	94.00	95.00 +1.0	10.00	17.86 +7.9	92.00	100.00 +8.0
DINOv3 (dino.txt)	44.00	92.31 +48.3	92.00	100.00 +8.0	36.00	64.29 +28.3	84.00	86.15 +2.2
SigLIP so400M	42.00	88.46 +46.5	94.00	97.50 +3.5	26.00	57.14 +31.1	86.00	95.38 +9.4
SigLIP 2 so400M	44.00	92.31 +48.3	92.00	95.00 +3.0	30.00	67.86 +37.9	86.00	92.31 +6.3
SigLIP 2 giant	44.00	100.00 +56.0	94.00	97.50 +3.5	34.00	78.57 +44.6	78.00	90.77 +12.8
DINOv3 (k-NN)	48.00	100.00 +52.0	94.00	100.00 +6.0	48.00	71.43 +23.4	74.00	63.08 -10.9
EfficienNetV2-XL	48.00	96.15 +48.2	92.00	100.00 +8.0	60.00	85.71 +25.7	62.00	55.38 -6.6
EfficienNet-L2	52.00	100.00 +48.0	96.00	100.00 +4.0	66.00	82.14 +16.1	74.00	60.00 -14.0
EVA-02	52.00	100.00 +48.0	96.00	100.00 +4.0	60.00	82.14 +22.1	74.00	60.00 -14.0

Table 9. Recall comparison between ImGT and WeaselGT. MLLMs (top) show large gains, often reaching perfect recall, indicating sensitivity to noisy or ambiguous ImGT labels. VLMs (center) also improve, reflecting reduced fine-grained sensitivity. Supervised models (bottom) sometimes drop, suggesting overfitting to ImGT. *Note: MLLMs recall is computed only for the CW setup.*

		ImGT	ReGT						
	Batch	A_{625}	A_{625}	S_{352}	$S{+}_{316}$	$S{-}_{36}$	M_{240}	$M{+}_{221}$	$M{-}_{19}$	OOP
LLaVA-OV	1	42.56	52.00	46.31	48.10	30.56	53.75	55.20	36.84	168
	5	34.88	42.56	37.78	40.19	16.67	41.67	43.44	21.05	233
	10	27.20	35.36	30.97	32.28	19.44	32.92	34.39	15.79	258
Qwen3-VL	1	64.96	73.12	69.32	73.10	36.11	75.00	77.83	42.11	28
	5	63.52	72.00	70.45	74.37	36.11	70.42	72.85	42.11	44
	10	64.16	72.16	71.88	75.95	36.11	68.75	72.40	26.32	61
GPT-4o	1	73.92	80.80	79.55	84.49	36.11	80.00	83.71	36.84	23
	5	74.40	81.76	80.40	84.81	41.67	81.25	85.07	36.84	27
	10	74.24	81.92	80.68	85.44	38.89	81.25	85.07	36.84	23

Table 10. Image batch size comparison for MLLMs on 625 sampled images (one per class) using batches of 1, 5, and 10. Qwen3-VL and GPT-4o show consistent performance across batch sizes, while LLaVA-OV accuracy drops with larger batches. Thus, batch size 10 is used for Qwen3-VL and GPT-4o, and 1 for LLaVA-OV.

		ImGT	ReGT	
Model	In-Batch Ordering	A_{6250}	A_{6250}	OOP
Qwen3-VL	Random	63.47	70.83	734
	Same-Class	76.96	78.19	470
GPT-4o	Random	75.78	80.82	324
	Same-Class	86.03	84.61	155

Table 11. We compare random class mixtures to same-class batches on 6250 images (10 per class). Grouping by class causes MLLMs to assign identical labels within a batch, inflating ImGT accuracy and agreement with ReGT. Random ordering reduces this bias and is used in all experiments with image-batched inputs.

Composition bias. Using this fixed image batch size for Qwen3-VL and GPT-4o, we evaluate the effect of in-batch image ordering on model behavior. We compare randomly mixed batches with batches containing only images from the same ImGT class. As shown in Tab. 11, class-grouped batches frequently cause MLLMs to assign the same label to every image, inflating ImGT and ReGT accuracy. Random ordering mitigates this batch-class bias; therefore, we adopt random ordering for all experiments with these models.

Class names or class IDs? With image batch size and ordering fixed, we compare two response formats: requiring the models to output a class ID from a provided mapping or directly output a Class Name from the supplied list. As shown in Tab. 13, GPT-4o performs similarly in both

	Img Pos.	Batch	ImGT $A_{63(62)}$	ReGT $A_{63(62)}$	OOP
LLaVA-OV	1st	1	44.44	60.32	18
	1st	5	42.86 -1.6	58.73 -1.6	20 +2
	1st	10	33.33 -11.1	52.38 -7.9	23 +5
	5th	1	55.56	69.84	11
	5th	5	26.98 -28.6	34.92 -34.9	28 +17
	5th	10	31.75 -23.8	39.68 -30.2	25 +14
	10th	1	37.10	50.00	22
	10th	5	16.13 -21.0	29.03 -21.0	38 +16
	10th	10	17.74 -19.4	30.65 -19.4	33 +11
Qwen3-VL	1st	1	63.49	73.02	1
	1st	5	66.67 +3.2	76.19 +3.2	2 +1
	1st	10	66.67 +3.2	76.19 +3.2	2 +1
	5th	1	71.43	85.71	0
	5th	5	68.25 -3.2	80.95 -4.8	5 +5
	5th	10	66.67 -4.8	77.78 -7.9	5 +5
	10th	1	61.29	72.58	7
	10th	5	53.23 -8.1	69.35 -3.2	8 +1
	10th	10	56.45 -4.8	70.97 -1.6	9 +2
GPT-4o	1st	1	69.84	79.37	2
	1st	5	73.02 +3.2	82.54 +3.2	2
	1st	10	69.84	79.37	1 -1
	5th	1	76.19	88.89	1
	5th	5	74.60 -1.6	88.89	3 +2
	5th	10	74.60 -1.6	85.71 -3.2	2 +1
	10th	1	70.97	83.87	4
	10th	5	70.97	85.48 +1.6	3 -1
	10th	10	80.65 +9.7	90.32 +6.5	3 -1

Table 12. Accuracy of the 1st, 5th, and 10th images from request batches of size 1, 5, and 10, evaluated on 625 sampled images (one per class). Image positions vary by batch size. Qwen3-VL and GPT-4o show stable accuracy across positions, while LLaVA-OV's accuracy declines for later images, consistent with its sensitivity to larger batches (see Tab. 10).

settings, although the Class Name format occasionally produces out-of-prompt (OOP) outputs that can later be leveraged in the CW+ experiment. For LLaVA-OV, InternVL3.5, and Qwen3-VL, the Class Name format yields noticeably higher accuracy compared to the ID format, while still providing OOP predictions. Therefore, we use the Class Name format in all experiments.

Language-models (LMs) comparison for mapping. We evaluate different LMs for our OW setup: (i) Sentence-BERT [31], which is optimized for general-purpose sentence-level semantic similarity and provides embeddings aligned for textual comparison (following a similar approach to our OW semantic-similarity baseline in Conti et al. [7]); (ii) SigLIP 2 [35], a state-of-the-art language model designed to produce high-quality, image-grounded text embeddings; and (iii) the newly released Qwen3 text encoder [41], tailored for multimodal models and naturally integrated with the Qwen3-VL architecture. The results are presented in Tab. 17. Templating improves performance for all encoders.

PaliGemma 2 and GPT-4o perform best when paired with the SigLIP 2, while LLaVA-OV, InternVL3.5 and Qwen3-VL achieve the highest performance with Qwen3-Embedding-8B. The model-specific LM selected based on OW accuracy is used for evaluation in both the OW and CW+ setups in the main paper.

Distractor choice in the MC setup We explore several methods for the informed selection of challenging distractors for a class c in the 4-choice MC setup: (i) using the confusion matrix of the EVA-02 model, and (ii) selecting classes closest to c in the BERT embedding space, similar to Liu et al. [21]. The results are shown in Tab. 18.

The BERT-based method can yield highly challenging distractors (*e.g.* distractors for "computer keyboard" include "typewriter keyboard", "keyboard space bar", and "laptop computer"), but it may also produce distractors that are no more difficult than random choices (*e.g.* the nearest classes to "lens cap" are "swimming cap", "bottle cap", and "shower cap"). Overall, BERT-based distractors are more challenging than completely random selection, yet remain less difficult than those derived from the EVA-02.

For example, in the pure ImGT with distractors setup (a common baseline for MLLMs image recognition), GPT-4o's performance on ImGT drops nearly twice as much with EVA-02 distractors vs. BERT ones (90.66% vs. 95.86%), compared to random distractors (99.62%).

D. Additional Results

D.1. Detailed overview of the OOP predictions

We present the distribution of out-of-prompt MLLMs predictions under the CW setup across four types in Tab. 15. We also report the exact number of OOP predictions for each label category, along with the number correctly mapped (using the best model-specific LM) in the CW+ setup, in Tab. 14. Predictions in category N do not require mapping, as any prediction for images in this category is considered correct in our setup.

D.2. Correlation of recall and correctness patterns

To analyze similarity in model behavior, we compute two complementary correlation measures, corresponding to the two panels in Fig. 9.

Model	Response	ImGT	ReGT							
		A_{625}	A_{625}	S_{352}	$S{+}_{316}$	$S{-}_{36}$	M_{240}	$M{+}_{221}$	$M{-}_{19}$	OOP
LLaVA-OV	ID	7.68	14.24	9.38	9.81	5.56	9.58	9.95	5.26	0
	Class name	42.56 +34.9	52.00 +37.8	46.31	48.10	30.56	53.75	55.20	36.84	168
InternVL3.5	ID	57.60	67.20	62.22	64.24	44.44	70.00	72.85	36.84	6
	Class name	63.68 +6.1	72.96 +5.8	67.05	69.30	47.22	77.92	80.54	47.37	6
Qwen3-VL	ID	48.00	57.28	54.55	56.96	33.33	55.42	58.37	21.05	0
	Class name	64.16 +16.2	72.16 +14.9	71.88	75.95	36.11	68.75	72.40	26.32	61
GPT-4o	ID	70.24	78.08	75.00	79.43	36.11	79.58	83.26	36.84	0
	Class name	74.24 +4.0	81.92 +3.8	80.68	85.44	38.89	81.25	85.07	36.84	23

Table 13. Comparison of two response formats for MLLMs on 625 sampled images (one per class): *ID*: predicting a class ID via a mapping (*e.g.* 0 – "tench", . . .) and *Class Name*: selecting directly from class names. Accuracy deltas relative to the ID format. While accuracy gains over the ID format vary by model, Class Name consistently performs better and can produce OOP outputs useful for CW+. We therefore use Class Name in all experiments.

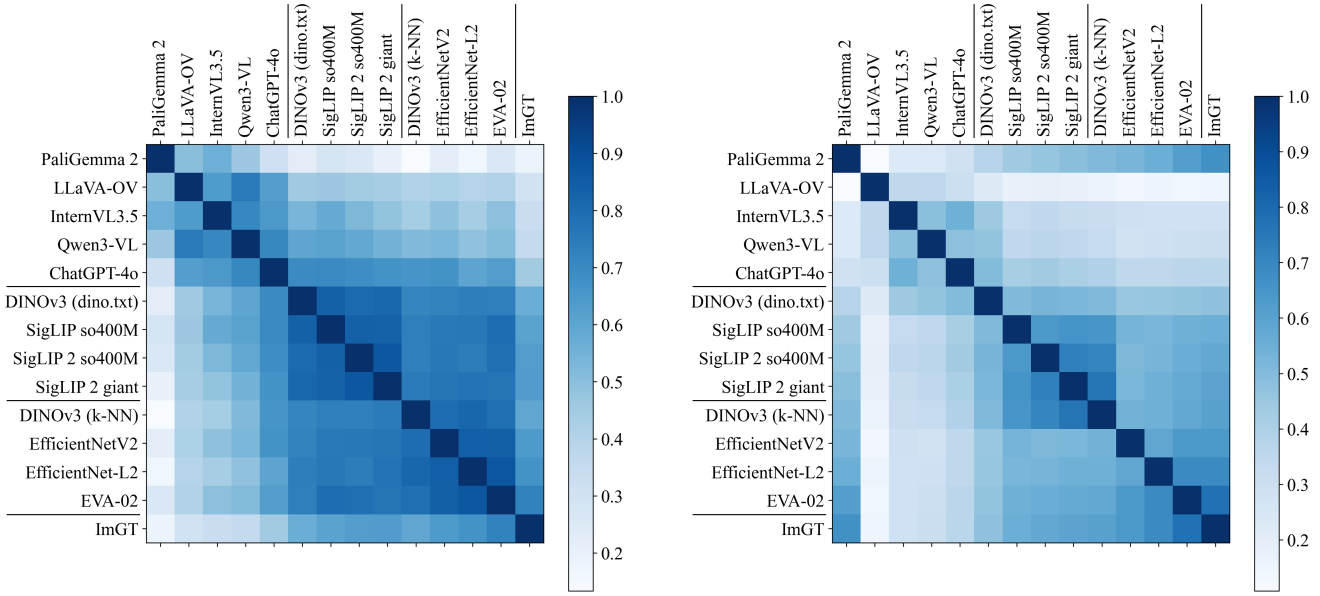


Figure 9. Correlation matrices across models evaluated on reannotated labels (ReGT). **Left:** Computes per-class recall for single-label images only and measures cross-model similarity using Spearman correlation. **Right:** Builds an image correctness vector including all images with valid label(s) and computes cross-model similarity using the Phi coefficient. The blocks indicate MLLMs, VLMs, supervised models and original ground-truth (ImGT).

Per-class recall correlation (left). For each model, we compute per-class recall for every class in the reannotated image subset, restricting the analysis to single-label images only. Although the subset was reannotated for 625 classes, the images span 665 ImageNet-1k classes. We calculate the Spearman correlation between these per-class recall vectors across models. This correlation matrix captures similarity in class-wise error patterns rather than raw accuracy.

A clear structure emerges according to the training paradigm. Supervised models (EfficientNetV2, EfficientNet-L2, EVA-02) and the k-NN variant of DINOv3 form a tight, highly correlated cluster. VLMs: DINOv3 (dino.txt), SigLIP, and SigLIP 2 are strongly aligned with each other and moderately correlated with supervised models. In contrast, MLLMs exhibit lower similarity to traditional vision models, indicating that their failure modes differ substantially from supervised and self-supervised approaches. We also compute per-class recall by treating ImGT as predictions. This vector shows strong alignment with high-capacity supervised and self-supervised models.

		OOP		Correctly Mapped	
		#	%	#	%
InternVL3.5	A_{31250}	271	0.87	83	30.63
	S_{18071}	145	0.80	34	23.45
	$S+_{16177}$	130	0.80	34	26.15
	$S-_{1894}$	15	0.79	0	0.00
	M_{11834}	92	0.78	15	16.30
	$M+_{10756}$	79	0.73	12	15.19
	$M-_{1078}$	13	1.21	3	23.08
	N_{1345}	34	2.53	–	–
GPT-4o	A_{31250}	1647	5.27	554	33.64
	S_{18071}	717	3.97	142	19.80
	$S+_{16177}$	553	3.42	119	21.52
	$S-_{1894}$	164	8.66	23	14.02
	M_{11834}	710	6.00	192	27.04
	$M+_{10756}$	603	5.61	177	29.35
	$M-_{1078}$	107	9.93	15	14.02
	N_{1345}	220	16.36	–	–
Qwen3-VL	A_{31250}	3290	10.53	1275	38.75
	S_{18071}	1671	9.25	497	29.74
	$S+_{16177}$	1410	8.72	456	32.34
	$S-_{1894}$	261	13.78	41	15.71
	M_{11834}	1294	10.93	453	35.01
	$M+_{10756}$	1135	10.55	421	37.09
	$M-_{1078}$	159	14.75	32	20.13
	N_{1345}	325	24.16	–	–
LLaVA-OV	A_{31250}	8364	26.76	3720	44.48
	S_{18071}	4355	24.10	1797	41.26
	$S+_{16177}$	3823	23.63	1679	43.92
	$S-_{1894}$	532	28.09	118	22.18
	M_{11834}	3437	29.04	1351	39.31
	$M+_{10756}$	3101	28.83	1282	41.34
	$M-_{1078}$	336	31.17	69	20.54
	N_{1345}	572	42.53	–	–

Table 14. Overview of MLLMs out-of-prompt across label categories. Counts (#) indicate predictions falling outside the provided class names in the CW setup and the number correctly mapped (#) in CW+. Harder label categories: M– and S–, which do not contain the ImGT label, and N, which contains no label at all – exhibit a higher ratio of OOP predictions.

Image-level correctness correlation (right). We also construct, for each model, a binary image-level correctness vector over all reannotated images with valid label(s), where each entry indicates whether the model prediction is correct. We compute pairwise similarity between these binary vectors using the Phi coefficient. This analysis operates at the image level and includes multi-label cases. The resulting matrix reflects agreement on which specific images are classified correctly or incorrectly. While the results differ slightly from the left panel, they still highlight the MLLMs cluster, which is weakly correlated with other models.

Model	Partial	ImageNet	Abstain	Wrong
InternVL3.5	21.77	0.0	1.11	77.12
GPT-4o	35.34	3.22	6.38	55.07
Qwen3-VL	26.81	1.67	0.0	71.52
LLaVA-OV	36.29	2.59	0.04	61.08

Table 15. Distribution of out-of-prompt MLLMs predictions. "Partial" denotes cases where the predicted string is a subset of any of the provided class names, regardless of word order. "ImageNet" corresponds to predictions that exactly match the commonly used OpenAI ImageNet-1k class names. "Abstain" refers to predictions in the form of "I don't know" or similar expressions (including LLM-generated variations). "Wrong" includes all remaining predictions.

D.3. Stability under repeated evaluation

The results of accuracy deviation are presented in Tab. 16. With the temperature set to 0, GPT-4o exhibits non-zero variability, whereas LLaVA-OV, InternVL3.5, and Qwen3-VL behave deterministically.

		ImGT	ReGT							
Model	Batch	A_{625}	A_{625}	S_{352}	$S+_{316}$	$S-_{36}$	M_{240}	$M+_{221}$	$M-_{19}$	OOP
GPT-4o	10	74.69 $^{\pm 0.19}$	81.32 $^{\pm 0.18}$	79.87 $^{\pm 0.19}$	84.49 $^{\pm 0.21}$	39.25 $^{\pm 0.82}$	80.89 $^{\pm 0.35}$	84.72 $^{\pm 0.36}$	36.33 $^{\pm 0.58}$	30.13 $^{\pm 1.11}$

Table 16. Accuracy deviation over 31 repeated evaluations of a 625 sampled images (one per class) using GPT-4o at temperature 0. Despite deterministic settings, small non-zero variability appears, shown by 95 % confidence intervals. LLaVA-OV, InternVL3.5, and Qwen3-VL are fully deterministic and omitted.

		ImGT	ReGT						
	Emb. Space	A_{31250}	A_{31250}	S_{18071}	$S+_{16177}$	$S-_{1894}$	M_{11834}	$M+_{10756}$	$M-_{1078}$
PaliGemma 2	Sentnce-BERT	30.60	41.41	32.58	34.01	20.38	48.24	49.58	34.88
	Sentence-BERT†	31.89	42.81	33.86	35.32	21.44	49.98	51.33	36.55
	Qwen3-Embedding-8B	33.24	43.53	35.49	37.05	22.23	49.37	50.99	33.21
	Qwen3-Embedding-8B†	36.52	47.51	39.05	40.71	**24.92**	54.45	56.16	37.38
	SigLIP 2	34.56	45.04	36.45	38.23	21.28	51.90	53.51	35.81
	SigLIP 2†	**37.11**	**47.94**	**39.74**	**41.49**	24.76	**54.55**	**56.24**	**37.66**
LLaVA-OV	Sentence-Bert	55.72	64.56	61.13	64.42	33.05	65.76	68.60	37.38
	Sentence-BERT†	56.57	65.60	61.76	65.03	33.84	67.54	70.43	38.78
	Qwen3-Embedding-8B	57.56	65.71	62.22	65.42	34.90	67.14	70.22	36.36
	Qwen3-Embedding-8B†	**62.00**	**70.58**	67.11	70.67	36.69	**72.52**	**75.87**	39.05
	SigLIP 2	59.40	67.24	64.58	68.12	34.42	67.57	70.59	37.38
	SigLIP 2†	61.98	70.35	**67.38**	**70.84**	**37.91**	71.50	74.67	**39.89**
InternVL3.5	Sentnce-BERT	51.94	61.15	55.51	58.48	30.15	65.33	68.34	35.25
	Sentence-SBERT†	53.11	62.63	56.75	59.71	31.47	67.36	70.30	38.03
	Qwen3-Embedding-8B	53.72	62.07	56.54	59.51	31.20	66.21	69.20	36.36
	Qwen3-Embedding-8B†	**59.23**	**68.18**	62.44	**65.78**	33.90	**73.33**	**76.66**	**40.07**
	SigLIP 2	55.25	63.09	58.67	61.85	31.47	65.65	68.70	35.25
	SigLIP 2†	58.91	67.16	**62.53**	65.76	**34.90**	70.51	73.72	38.50
Qwen3-VL	Sentence-BERT	57.65	66.51	62.72	66.22	32.84	68.49	71.46	38.87
	Sentence-BERT†	60.35	69.21	65.41	69.05	34.32	71.51	74.65	40.26
	Qwen3-Embedding-8B	59.28	66.89	62.45	65.96	32.42	69.90	73.21	36.83
	Qwen3-Embedding-8B†	**68.74**	**76.68**	**73.61**	**77.79**	**37.91**	**78.71**	**82.46**	**41.28**
	SigLIP 2	59.27	65.66	63.86	67.96	28.83	64.50	67.83	31.26
	SigLIP 2†	65.82	73.07	71.23	75.40	35.59	72.82	76.40	37.01
GPT-4o	Sentence-BERT	64.14	71.49	70.22	74.57	33.05	70.20	73.77	34.51
	Sentence-BERT†	64.84	72.58	71.03	75.28	34.69	71.83	75.51	35.06
	Qwen3-Embedding-8B	63.91	70.59	68.62	72.72	33.63	70.25	74.00	32.84
	Qwen3-Embedding-8B†	70.32	77.29	75.61	80.23	36.22	**77.28**	**81.41**	**36.09**
	SigLIP 2 ViT-gopt-384	67.42	73.94	73.11	77.41	36.38	72.26	76.20	32.93
	SigLIP 2† ViT-gopt-384	**70.92**	**77.58**	**76.59**	**81.08**	**38.23**	76.55	80.75	34.69

Table 17. Results for the OW setup with different language models (LMs) used for prediction mapping are reported. † denotes the use of templating following Radford et al. [29]. Sentence-BERT is included, as it is used for semantic similarity in Conti et al. [7]. Templating improves performance for all LMs. SigLIP 2 encoding performs best for PaliGemma 2 and GPT-4o, while Qwen3-Embedding-8B encoding achieves the highest performance for LLaVA-OV, InternVL3.5, and Qwen3-VL. The model-specific LM selected based on OW accuracy is used for evaluation in both the OW and CW+ setups in the main paper.

		ImGT	ReGT					
	Distractors	A_{625}	A_{625}	S_{352}	$S{+}_{316}$	$S{-}_{36}$	M_{240}	$M{+}_{221}$	$M{-}_{19}$
LLaVA-OV	ImGT + random	99.28 ±0.15	90.75 ±0.11	89.36 ±0.13	99.52 ±0.14	0.18 ±0.25	91.52 ±0.18	99.39 ±0.20	0.00 ±0.00
	ImGT + `confEVA(ImGT)`	79.93 ±0.32	76.02 ±0.30	73.13 ±0.45	81.46 ±0.50	0.00 ±0.00	76.96 ±0.47	83.58 ±0.51	0.00 ±0.00
	ImGT + `confBERT(ImGT)`	88.59 ±0.29	82.65 ±0.26	79.66 ±0.42	88.73 ±0.46	0.00 ±0.00	84.65 ±0.32	91.93 ±0.35	0.00 ±0.00
	ReGT + `confEVA(ReGT)`	44.28 ±0.26	78.36 ±0.39	81.42 ±0.43	81.49 ±0.48	80.83 ±1.40	70.90 ±0.69	71.77 ±0.64	60.78 ±3.86
	ReGT + `confBERT(ReGT)`	49.19 ±0.25	84.92 ±0.42	89.08 ±0.42	89.21 ±0.47	87.99 ±1.24	76.75 ±0.82	76.57 ±0.91	78.78 ±2.66
	ImGT + ReGT + random	89.99 ±0.28	95.21 ±0.17	95.44 ±0.21	99.55 ±0.14	59.32 ±1.38	94.22 ±0.27	99.71 ±0.16	30.39 ±2.89
	ImGT + ReGT + `confEVA(ImGT,ReGT)`	82.17 ±0.31	90.01 ±0.20	87.78 ±0.23	90.74 ±0.25	61.74 ±1.08	91.92 ±0.31	97.23 ±0.25	30.22 ±2.95
	ImGT + 999 (Closed World (CW))	42.56 ±0.00	52.00 ±0.00	46.31 ±0.00	48.10 ±0.00	30.56 ±0.00	53.75 ±0.00	55.20 ±0.00	36.84 ±0.00
InternVL3.5	ImGT + random	99.64 ±0.11	90.97 ±0.08	89.54 ±0.10	99.72 ±0.11	0.18 ±0.25	91.83 ±0.13	99.72 ±0.14	0.00 ±0.00
	ImGT + `confEVA(ImGT)`	83.72 ±0.30	78.95 ±0.24	76.04 ±0.34	84.71 ±0.38	0.00 ±0.00	80.31 ±0.40	87.21 ±0.44	0.00 ±0.00
	ImGT + `confBERT(ImGT)`	92.68 ±0.22	85.42 ±0.18	82.64 ±0.26	92.06 ±0.29	0.00 ±0.00	87.50 ±0.28	95.02 ±0.30	0.00 ±0.00
	ReGT + `confEVA(ReGT)`	46.49 ±0.19	82.83 ±0.37	84.41 ±0.33	84.71 ±0.38	81.81 ±0.82	78.14 ±0.75	79.10 ±0.65	67.06 ±3.56
	ReGT + `confBERT(ReGT)`	51.32 ±0.19	88.97 ±0.24	91.84 ±0.26	92.06 ±0.29	89.88 ±0.62	83.24 ±0.54	83.35 ±0.53	82.00 ±2.58
	ImGT + ReGT + random	91.98 ±0.22	94.90 ±0.15	94.99 ±0.16	99.72 ±0.11	53.41 ±1.20	94.07 ±0.30	99.83 ±0.10	27.17 ±3.60
	ImGT + ReGT + `confEVA(ImGT,ReGT)`	85.89 ±0.29	90.53 ±0.24	88.78 ±0.38	92.86 ±0.36	52.96 ±1.76	91.79 ±0.36	97.35 ±0.23	27.17 ±2.91
	ImGT + 999 (Closed World (CW))	63.68 ±0.00	72.96 ±0.00	67.05 ±0.00	69.30 ±0.00	47.22 ±0.00	77.92 ±0.00	80.54 ±0.00	47.37 ±0.00
Qwen3-VL	ImGT + random	99.34 ±0.09	90.67 ±0.08	89.41 ±0.08	99.59 ±0.09	0.09 ±0.18	91.22 ±0.17	99.07 ±0.18	0.00 ±0.00
	ImGT + `confEVA(ImGT)`	85.80 ±0.29	80.74 ±0.26	77.67 ±0.27	86.52 ±0.31	0.00 ±0.00	82.59 ±0.42	89.70 ±0.46	0.00 ±0.00
	ImGT + `confBERT(ImGT)`	93.13 ±0.22	86.54 ±0.17	84.41 ±0.27	94.03 ±0.30	0.00 ±0.00	87.81 ±0.23	95.36 ±0.25	0.00 ±0.00
	ReGT + `confEVA(ReGT)`	47.14 ±0.17	82.01 ±0.36	86.40 ±0.30	86.52 ±0.31	85.39 ±1.26	73.09 ±0.77	73.92 ±0.67	63.50 ±3.69
	ReGT + `confBERT(ReGT)`	51.88 ±0.18	88.57 ±0.34	93.70 ±0.32	94.03 ±0.30	90.77 ±1.13	79.48 ±0.66	79.39 ±0.64	80.47 ±2.78
	ImGT + ReGT + random	91.95 ±0.25	94.23 ±0.11	94.12 ±0.14	99.59 ±0.09	46.06 ±1.46	93.60 ±0.21	99.64 ±0.11	23.43 ±2.48
	ImGT + ReGT + `confEVA(ImGT,ReGT)`	86.32 ±0.26	90.77 ±0.21	89.14 ±0.27	93.86 ±0.28	47.67 ±1.18	91.88 ±0.28	97.65 ±0.28	24.79 ±2.55
	ImGT + 999 (Closed World (CW))	64.16 ±0.00	72.16 ±0.00	71.88 ±0.00	75.95 ±0.00	36.11 ±0.00	68.75 ±0.00	72.40 ±0.00	26.32 ±0.00
GPT-4o	ImGT + random	99.62 ±0.07	90.95 ±0.05	89.49 ±0.07	99.67 ±0.07	0.09 ±0.18	91.85 ±0.10	99.75 ±0.11	0.00 ±0.00
	ImGT + `confEVA(ImGT)`	90.66 ±0.22	85.12 ±0.21	84.27 ±0.26	93.87 ±0.29	0.00 ±0.00	84.31 ±0.39	91.56 ±0.42	0.00 ±0.00
	ImGT + `confBERT(ImGT)`	95.86 ±0.17	88.76 ±0.12	87.71 ±0.19	97.70 ±0.21	0.00 ±0.00	88.75 ±0.18	96.38 ±0.20	0.00 ±0.00
	ReGT + `confEVA(ReGT)`	51.35 ±0.21	84.26 ±0.37	92.45 ±0.36	93.90 ±0.31	79.66 ±1.50	70.09 ±0.89	70.78 ±0.83	62.14 ±3.36
	ReGT + `confBERT(ReGT)`	54.10 ±0.12	89.37 ±0.35	96.72 ±0.23	97.75 ±0.20	87.63 ±1.11	77.14 ±0.76	76.91 ±0.77	79.80 ±2.69
	ImGT + ReGT + random	91.77 ±0.26	94.89 ±0.13	94.79 ±0.16	99.67 ±0.07	51.97 ±1.35	94.31 ±0.27	99.80 ±0.09	30.56 ±2.97
	ImGT + ReGT + `confEVA(ImGT,ReGT)`	88.31 ±0.29	92.52 ±0.15	91.59 ±0.20	96.36 ±0.15	49.73 ±1.27	92.86 ±0.31	98.16 ±0.20	31.24 ±3.30
	ImGT + 999 (Closed World (CW))	74.69 ±0.19	81.32 ±0.18	79.87 ±0.19	84.49 ±0.21	39.25 ±0.82	80.89 ±0.35	84.72 ±0.36	36.33 ±0.58

Table 18. 4-way multiple-choice (MC) results across 625 sampled images (one per class), reported with 95 % confidence intervals. `confEVA()` and `confBERT()` denote functions that return challenging distractors based on the confusion matrix of EVA-02 and the distance between class names in the BERT embedding space, respectively. The Closed World result corresponds to using 999 distractors.

GPT-4o Prompt

```
You are an image classifier.
You will receive up to 50 images in order (image "1" = first, "2" = second, etc.).
You are also provided with a list of class names:  {class_list}.

Classification Rules:
- For each image, return the single class name that best represents the main subject of the image.
- Choose only one class per image - the most relevant or dominant one.
- Only return classes from the provided list.

Output Rules:
- Return exactly one output per image.
- Each output must be only a single class name (no separators or lists).
- Do not include explanations, confidence scores, or extra text.
```

LLaVA-OV & InternVL3.5 & Qwen3-VL Prompt

```
You are an image classifier.
You will receive up to 50 images in order (image "1" = first, "2" = second, etc.).
You are also provided with a list of class names:  {class_list}.

Your output will be automatically structured as JSON with keys "1", "2", "3", etc.  corresponding
to the order of images in the request.
Each value should be the predicted class name for that image.

Classification Rules:
- For each image, return the class name only from the provided list.
- Only return classes from the provided list.

Output Rules:
- Return exactly one JSON key per image ("1", "2", "3", etc.).
- Each value must be only class names.
- Do not include explanations, confidence scores, or extra text.
```

Table 19. Prompts for the Closed World setup.

GPT-4o Prompt

```
You are an open-set fine-grained image classifier.
You will receive up to 50 images in order (image "1" = first, "2" = second, etc.).

Classification Rules:
- For each image, identify the dominant object.
- Return the most fine-grained, specific label that accurately describes that object (e.g.,
"golden retriever puppy", "1950s red convertible", "blue morpho butterfly", "ceramic coffee mug
with floral pattern").
- Use natural-language labels that reflect detailed visual distinctions such as species,
make/model, style, color, or material.
- Avoid generic terms like "dog", "car", or "bird" when a more specific subtype or description is
visually inferable.
- If the dominant object cannot be clearly identified, return a concise descriptive label of its
appearance (e.g., "abstract metal sculpture", "blurry human silhouette").
- Focus only on the dominant object, even if multiple are present.

Output Rules:
- Return exactly one output per image.
- The output must contain only the final label (no punctuation beyond normal text, no
explanations, confidence scores, or extra text).
```

LLaVA-OV & InternVL3.5 & Qwen3-VL Prompt

```
You are an open-set fine-grained image classifier.
You will receive up to 50 images in order (image "1" = first, "2" = second, etc.).

Your output will be automatically structured as JSON with keys "1", "2", "3", etc.  corresponding
to the order of images in the request.
Each value should be the predicted label for that image.

Classification Rules:
- For each image, identify the dominant object.
- Return the most fine-grained, specific label that accurately describes that object (e.g.,
"golden retriever puppy", "1950s red convertible", "blue morpho butterfly", "ceramic coffee mug
with floral pattern").
- Use natural-language labels that reflect detailed visual distinctions such as species,
make/model, style, color, or material.
- Avoid generic terms like "dog", "car", or "bird" when a more specific subtype or description is
visually inferable.
- If the dominant object cannot be clearly identified, return a concise descriptive label of its
appearance (e.g., "abstract metal sculpture", "blurry human silhouette").
- Focus only on the dominant object, even if multiple are present.

Output Rules:
- Return exactly one output per image.
- The output must contain only the final label (no punctuation beyond normal text, no
explanations, confidence scores, or extra text).
```

Table 20. Prompts for the Open World setup.

LLaVA-OV & InternVL3.5 & Qwen3-VL & GPT-4o Prompt

```
You are an image classifier.  You will receive one image.  You are also provided with four
multiple-choice options (A, B, C, D).

What is the main object in this image?  {dynamic_choices}

Classification Rules:
- For the image, return the letter (A, B, C, or D) that corresponds to the correct option.
- Only return one letter.

Output Rules:
- Return exactly one letter (A, B, C, or D).
- Do not include explanations, the class name, or extra text.
- Your answer must be only the letter.
```

Table 21. Prompt for the Multiple-Choice setup. The same prompt is used for all MLLMs.

620 – laptop computer

681 – notebook computer

In the modern context, these terms effectively mean the same thing. Distinguishing between them solely from visual features is close to impossible. The image content of the classes is the same.

836 – sunglass

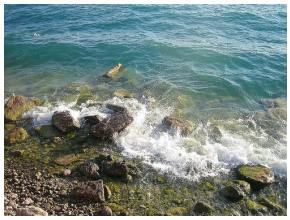
837 – sunglasses

Sunglass is defined as "a convex lens that focuses the rays of the sun; used to start a fire" in WordNet, the difference was lost in the original annotation process. The image content of the classes is the same.

742 – printer

713 – photocopier

Printers and photocopiers used to be separate devices, but there are also multi-functional ones. Distinguishing them visually is difficult, and in modern context, stand-alone photocopiers are no longer commonly used.

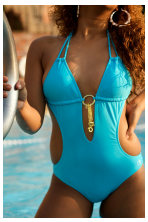


975 – lakeshore

978 – seashore

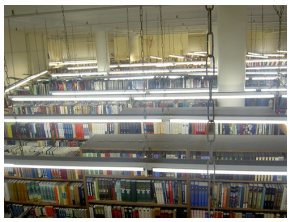
It's difficult to identify whether the water comes from ocean, sea, lake or river, unless it is a well-known geographical location or a characteristic wildlife can be observed.

638 – tights/leotard



639 – bathing suit

The original meaning of the first term was lost during annotation, similar to what happened with "sunglass". As a result, the image content of the classes is now indistinguishable based on visual features.



454 – bookstore

624 – library

It is challenging to distinguish a bookstore from a library solely from an interior view, unless a distinguishing object like a cash register is visible. This leaves room for ambiguity and speculation.

Figure 10. Class pairs (part 1/2) considered equivalent during evaluation (*i.e.* for a pair of classes $\{c_1, c_2\}$ and an image labeled c_1, a predicted label c_2 is also considered correct). Each pair is accompanied by a brief note explaining the decision.

657 – missile

744 – projectile, missile

The image content of the classes is mostly the same: the majority of the images are missiles.

461 – breastplate

524 – cuirass

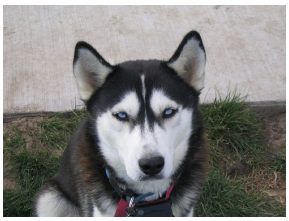
Breastplate only provides front coverage, while a cuirass covers the whole body. They share the same visual features from the front view, which is the predominant one.

435 – bathtub

876 – tub, vat

The image content of the classes is mostly the same.

248 – Eskimo dog



250 – Siberian Husky

The majority of the "Eskimo dog" class are huskies.

482 – cassette player

848 – tape player

Distinguishing between a cassette player and tape player from images is often unreliable. The terms are commonly used interchangeably, as the visual differences are minimal.

899 – water jug

725 – drink pitcher

Both classes represent containers for pouring liquids with identical structures; it is hard to tell when a water jug becomes a pitcher and vice versa.

Figure 11. Class pairs (part 2/2) considered equivalent during evaluation (*i.e.* for a pair of classes $\{c_1, c_2\}$ and an image labeled c_1, a predicted label c_2 is also considered correct). Each pair is accompanied by a brief note explaining the decision.