Indexing Multimodal Language Models for Large-scale Image Retrieval –supplementary material–

Bahey Tharwat Giorgos Kordopatis-Zilos Pavel Suma Ian Reid Giorgos Tolias

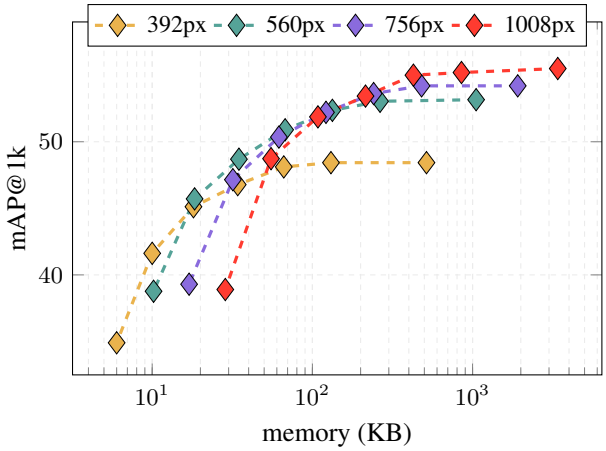

Figure A. **Impact of compression for different resolutions.** mAP@1k of Qwen on ILIAS for seven PQ compressions (*i.e.* PQ {1, 4, 8, 16, 32, 64, 128} from right to left) and four image resolutions (*i.e.* {392px, 560px, 756px, 1008px} in different colors).

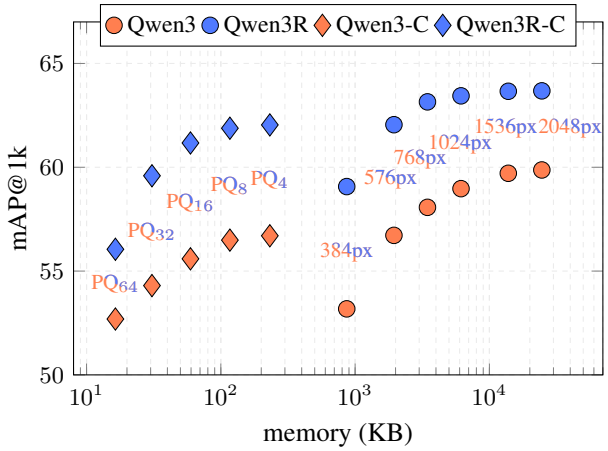

Figure B. **Impact of compression and resolution on a better MLLM.** mAP@1k comparison of Qwen3R and Qwen3 on ILIAS. Compression is applied via PQ with 560px image resolution.

A. Additional results

Impact of compression for different resolutions. We investigate how input image resolution affects performance when PQ is applied with varying compression levels. Fig. A shows the results for seven PQ compression rates and four input image resolutions. For a larger memory footprint, higher-resolution images consistently outperform lower-resolution ones when compared at equivalent memory requirements. However, this trend does not hold for a smaller memory footprint. In particular, PQ_{128} shows substantial performance degradation across all resolutions. Consequently, within the 10KB to 100KB memory range, a resolution of 560px yields the best performance; whereas, in the extremely low-memory regime, *i.e.* <10KB, a resolution of 392px performs best. We conclude that aggressive PQ compression severely compromises performance. Beyond a certain compression level, it becomes more effective to reduce memory usage by lowering the input resolution rather than increasing PQ compression further.

Impact of compression and resolution on better MLLMs. To assess the effectiveness of our compression strategy and the transferability of our findings to more recent MLLM architectures, we evaluate Qwen3R and Qwen3 under different PQ compression levels and image resolutions. We choose image resolutions that are divisible by the employed ViT's patch size. Fig. B shows the results on ILIAS. Qwen3R consistently outperforms Qwen3, highlighting the effectiveness of task-specific model adjustments and large-scale fine-tuning. Compared with the results from the main paper, both models consistently outperform their predecessor, Qwen, across all memory footprints.

Performance comparison over different semantic categories and domains. In Fig. C, we compare methods by averaging query performance across the mid-level categories defined in the ILIAS taxonomy. AMES outperforms the LLM-based approaches only in the landmark domain, particularly in the architecture category, which aligns with its training domain, indicating a strong domain bias. Although AMES generally improves upon global across most categories, there are some exceptions, *e.g.* jewelry, footwear, or textiles, presumably due to their semantic distance from its training domain. On the contrary, Qwen-based re-ranking consistently improves performance across all categories, showcasing the value of large-scale and diverse domain training. Interestingly, LamRA only surpasses Qwen on a few categories, showcasing once more that its generic training compromises performance on the specific task of instance-level retrieval, which is the focus of this work.

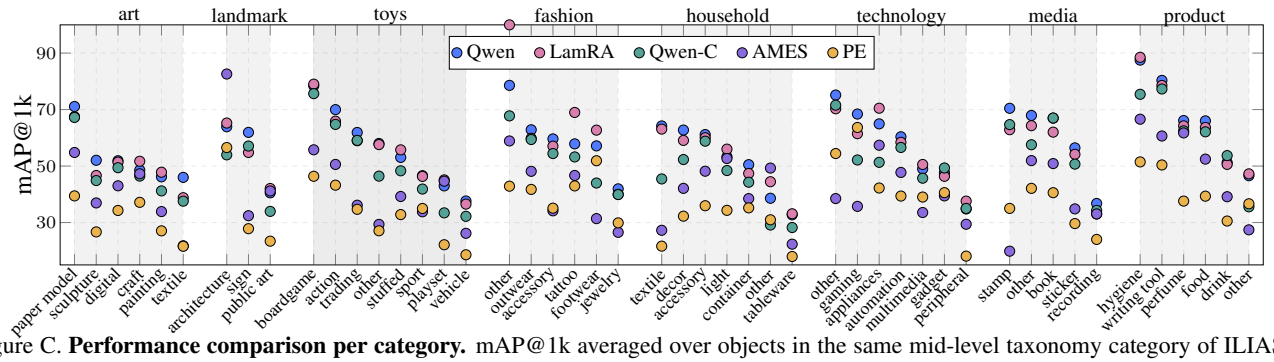

Figure C. **Performance comparison per category.** mAP@1k averaged over objects in the same mid-level taxonomy category of ILIAS, grouped by their primary-level category size, with sorting within each group by Qwen performance. Comparison between Qwen with and without compression (Qwen-C), AMES, LamRA, and PE.

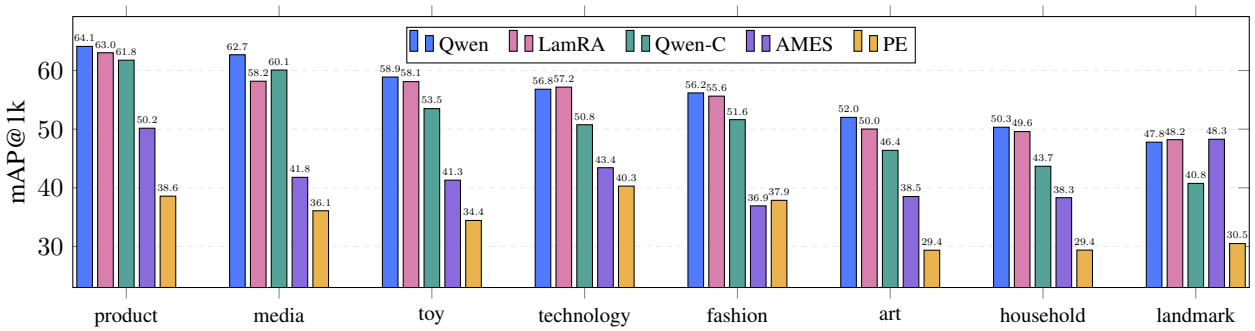

Figure D. **Performance comparison per domain.** mAP@1k averaged over objects in the same coarser taxonomy level of ILIAS, sorted by Qwen performance. Comparison between Qwen without and with compression (Qwen-C), AMES, LamRA, and PE.

Model	392px	560px	756px
No-rerank	33.4	33.4	33.4
Qwen-3B	19.7	24.3	25.3
Qwen-7B	48.5	53.2	54.4
Qwen-32B	48.1	51.3	52.0
Qwen-72B	54.2	54.5	OOM

Table A. **Impact of MLLM size.** mAP@1k of four Qwen variants with three image resolutions on ILIAS. No re-ranking is provided for comparison. OOM stands for out-of-memory.

Additionally, Fig. D displays the aggregated performance on the ILIAS domains according to the coarser taxonomy level. Qwen consistently outperforms all other models in most domains, demonstrating robust generalization in different object types. As in the subdomains, AMES achieves the best performance on landmarks, while LamRA slightly outperforms Qwen on technology.

Impact of MLLM size. Tab. A reports the results of Qwen in various sizes with different image resolutions. The largest models achieve the best performance. In contrast, the smallest variant performs notably worse than the no re-ranking baseline, suggesting that the fine-grained nature of the task requires sufficient model capacity. Nevertheless, model size alone does not fully explain performance trends; scaling from 7B to 32B does not bring a proportional boost. This is consistent for all image resolutions.

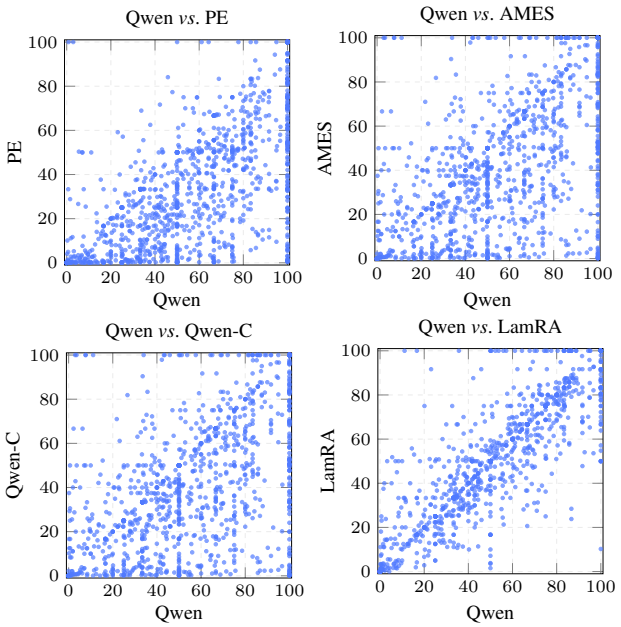

Figure E. **In-depth performance comparison.** AP per query comparison between Qwen and PE, AMES, Qwen-C, and LamRA. Each point corresponds to one query.

In-depth performance comparison. Fig. E presents scatterplots of AP per query, comparing Qwen with four other approaches. The superiority of Qwen over PE and AMES is evident, as the majority of points lie in the bottom-right region, indicating a higher AP for Qwen. Nonetheless,

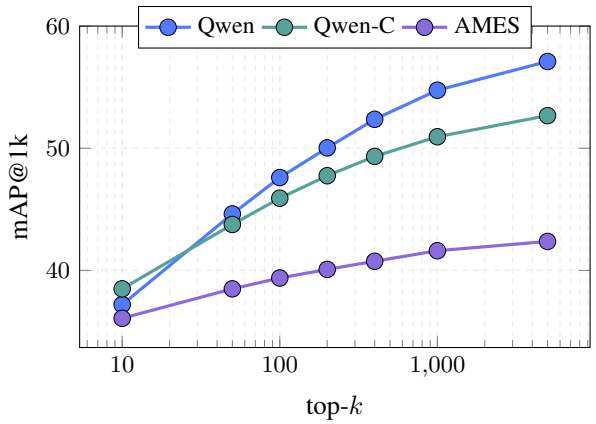

Figure F. **Impact of re-ranking.** mAP@1k of three models on ILIAS for increasing shortlist sizes, up to 5k images per query.

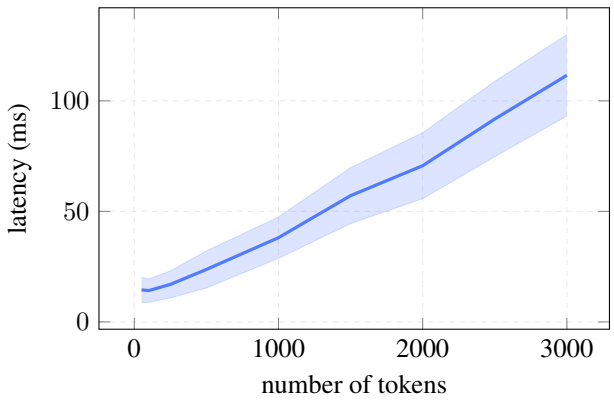

Figure G. **Latency *vs*. prompt length.** The plot shows the latency of the LLM as a function of the number of prompt tokens. The central line represents the mean latency, while the upper and lower bounds indicate the variance.

for PE, performance in several queries is degraded after re-ranking. Also, there are several queries where larger boosts are demonstrated with AMES. Similar observations can be inferred from the comparison with Qwen-C. Qwen and LamRA exhibit quite correlated performance, as reflected by the strong concentration of points along the diagonal.

Increasing the number of re-ranked images. Fig. F demonstrates re-ranking for different top-k, going as low as 10 and up to 5k shortlist images. Increasing k generally leads to improved performance for both Qwen models. Notably, we observe a performance crossover: while Qwen-C yields higher performance when re-ranking fewer candidates ($k < 50$), the standard Qwen model benefits significantly more from a larger candidate list, continuing to improve up to $k = 5000$. In contrast, AMES saturates early, suggesting that it struggles to effectively distinguish increasing numbers of hard negatives.

Latency vs. Prompt length. We analyze how the length of the prompt affects the processing time, measuring the latency of the LLM as a function of the input prompt length. Only the LLM component is considered in this measurement, as the representations extracted by the vision encoder are considered pre-computed. As shown in Fig. G, latency increases proportionally with the number of prompt tokens, indicating that longer textual inputs directly contribute to a higher computational overhead. In our case, the total prompt length is approximately 720 and 1200 tokens for image resolutions of 560px and 756px, respectively.

Additional qualitative results. In Fig. I, we provide additional qualitative examples. We draw similar conclusions to the ones from the main paper.

B. Dataset details

Below is the information regarding the datasets we used in our evaluation:

ILIAS [1] is a large-scale, multi-domain dataset designed for instance-level image retrieval. It consists of 1,000 object instances spanning various domains, based on which 1,232 queries and a database of 4,715 positives have been collected and combined with 100 million distractors sourced from YFCC100M. We adopt ILIAS as the primary testbed in our evaluation due to its large scale, diversity, and challenging nature.

INSTRE [6] is another instance-level multi-domain dataset. It consists of 200 objects and 1,250 single- and multi-object queries, and a database of 27.3k images.

$\mathcal{R}$***OP+1M*** [5] is the combination of two instance-level datasets of the landmark domain, *i.e.* $\mathcal{R}$Oxford [3] and $\mathcal{R}$Paris [4] containing 70 queries each from 11 landmarks, and databases of 5k and 6k, respectively. They are extended with one million distractors. We measure performance based on mAP on the *Medium* and *Hard* settings.

Product1M [7] is an instance-level dataset for product retrieval. It consists of 6.2k queries and 38.7k database images. In our evaluation, we randomly sample 1k queries.

C. Prompts

We provide a set of prompts that are tailored to the nature of the task and the domain of the dataset. All prompts are illustrated in Fig. H. Specifically, we include: (i) A generic prompt, applicable to arbitrary images and object types, and represent the generic case. Variants of such prompts are typically employed for generic retrieval tasks [2]. (ii) An object prompt, suited for datasets containing diverse objects. We use this prompt for ILIAS, INSTRE, and Prod-

```
You are given two images: a query and
a candidate. Determine whether the
candidate is similar to the query image.
Output strictly a single digit:
• 0 = the object instance does not appear.
• 1 = the object instance appears in the
  candidate.
Do not output anything else.
```

(a) `generic` prompt

```
You are given two images: a query and a
candidate. Determine whether the exact
same object instance from the query image
is present in the candidate image.
• The instance must be the same, not just
  a similar object.
• The instance may appear at a different
  scale, partially occluded, or among
  other objects.
Output strictly a single digit:
• 0 = the object instance does not appear.
• 1 = the object instance appears in the
  candidate.
Do not output anything else.
```

(b) `object` prompt

```
You are given two images: a query and a
candidate. Determine whether the exact
same landmark, building, or architectural
detail from the query image is present in
the candidate image.
• The instance must be the same, not just
  a similar-looking building or structure.
• The query image may show the entire
  landmark or just a specific, cropped
  part of it (like a doorway, statue, or
  window).
• The instance in the candidate image
  may appear at a different scale, from
  a different viewpoint/angle, under
  different lighting, or be partially
  occluded.
Output strictly a single digit:
• 0 = the object instance does not appear.
• 1 = the object instance appears in the
  candidate.
Do not output anything else.
```

(c) `landmark` prompt

Figure H. **Prompts for similarity estimation.** We use one generic and three task-specific prompts to evaluate the benchmarked MLLMs. We use `object` prompt for ILIAS, INSTRE, and Product1M, and the `landmark` prompt for $\mathcal{R}$OP+1M.

prompt	ILIAS	INSTRE	ROP	Prod1M
`generic`	42.0	94.6	63.0	72.3
`object`	**53.3**	**96.4**	65.7	**74.5**
`landmark`	40.0	90.7	**68.1**	66.9

Table B. **Prompt Sensitivity.** Retrieval performance (mAP) across datasets for different prompt types.

Model	Qwen3R	Qwen3R-C
`default`	60.2	58.8
`object`	**62.0**	**61.2**

Table C. **Qwen3R Prompting.** Retrieval performance (mAP) for Qwen3R and Qwen3R-C with the original generic `default` prompt, provided in the original repository, and with our task-specific `object` prompt.

uct1M. Regarding the latter, we also try prompts tailored for the product domain, but with no or insignificant improvements. (iii) A landmark prompt, designed for scenes involving buildings, monuments, or architectural elements, used in $\mathcal{R}$OP+1M.

We also provide an analysis of how MLLMs are sensitive to prompting. Tab. B shows that prompt sensitivity matters primarily at the semantic level. Task-specific prompts consistently perform best for the corresponding dataset, while unrelated domain-specific prompts perform poorly. At the same time, once the prompt is aligned with the task, its exact wording has only a limited effect: the task-specific prompt achieves 54.4 mAP, while five paraphrased variants remain close with a variance of $\sigma^2 = 0.585$, indicating that the MLLM is largely insensitive to prompt phrasing. Tab. C further shows that applying our task-specific prompt to Qwen3R consistently outperforms its default generic prompt, which comes with the Qwen3-Reranker model family, demonstrating the benefit of explicit task-aware prompting even for Qwen3R.

D. Transform details

We use a set of transformations designed to test a wide range of visual challenges, including photometric distortions, geometric and contextual changes, providing a comprehensive robustness evaluation across different retrieval approaches. Fig. J shows examples of the employed transformations. We always apply 20px zero-padding to prevent trivial self-similarity. The following is the list of transformations used in our robustness analysis:

- **contrast:** adjust the image contrast using a scaling factor in the range $[0.05, 20]$, where values below 1 decrease the contrast and values above 1 enhance it.
- **brightness:** adjust the luminance of the image by adding or subtracting a brightness offset in the range $[0.05, 20]$.
- **rotation:** rotate the image by an angle from $0°$ to $180°$, maintaining the image center.
- **downscale:** resize the image by scale factors from 0.5 to 0.05, simulating extreme resolution loss.

- **scale-bg:** scales the object down and places it over a random background, with object-to-background ratios ranging from 0 to 1.
- **blur:** applies Gaussian blur with kernel sizes increasing from $\sigma = 1$ to $\sigma = 15$.
- **tiling:** inserts random distractor patches from another image, covering one patch of $1/6$ of the image area to the total area of the image.
- **noise:** adds Gaussian noise with standard deviation σ varying from 0 to 1.0.
- **clutter:** similar to tiling, but it replaces the original background with a random scene. The object is merged onto this new background with an increase in the clutter density by adding 1 to 28 patches from another image.
- **occlusion:** apply circular black occluders that cover from 0% to 100% of the original image.

References

[1] Giorgos Kordopatis-Zilos, Vladan Stojnić, Anna Manko, Pavel Šuma, Nikolaos-Antonios Ypsilantis, Nikos Efthymiadis, Zakaria Laskar, Jiří Matas, Ondřej Chum, and Giorgos Tolias. ILIAS: Instance-level image retrieval at scale. In *CVPR*, 2025. 3

[2] Yikun Liu, Yajie Zhang, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiangchao Yao, Yanfeng Wang, and Weidi Xie. LamRA: Large multimodal model as your advanced retrieval assistant. In *CVPR*, 2025. 3

[3] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 3

[4] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 3

[5] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 3

[6] Shuang Wang and Shuqiang Jiang. INSTRE: A new benchmark for instance-level object retrieval and recognition. *TOMM*, 2015. 3

[7] Xunlin Zhan, Yangxin Wu, Xiao Dong, Yunchao Wei, Minlong Lu, Yichi Zhang, Hang Xu, and Xiaodan Liang. Product1m: Towards weakly supervised instance-level product retrieval via cross-modal pretraining. In *ICCV*, 2021. 3

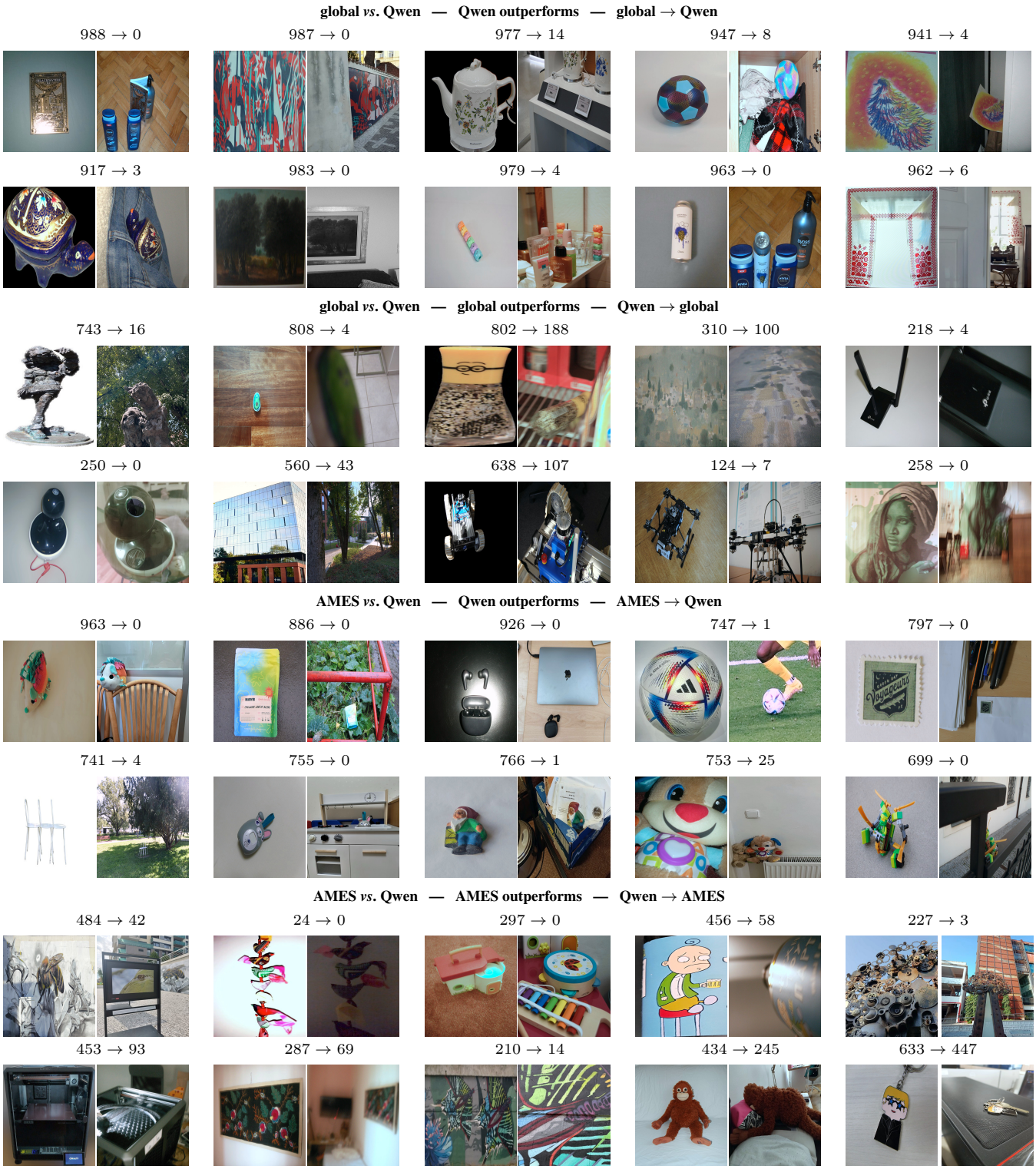

Figure I. **More qualitative examples where one method benefits the most compared to another.** We compare global (PE) and AMES *vs*. Qwen by showing pairs of query and positive image. → indicates the number of negative images ranked before the positive for two models, and it goes from the weaker to the stronger model for each pair.

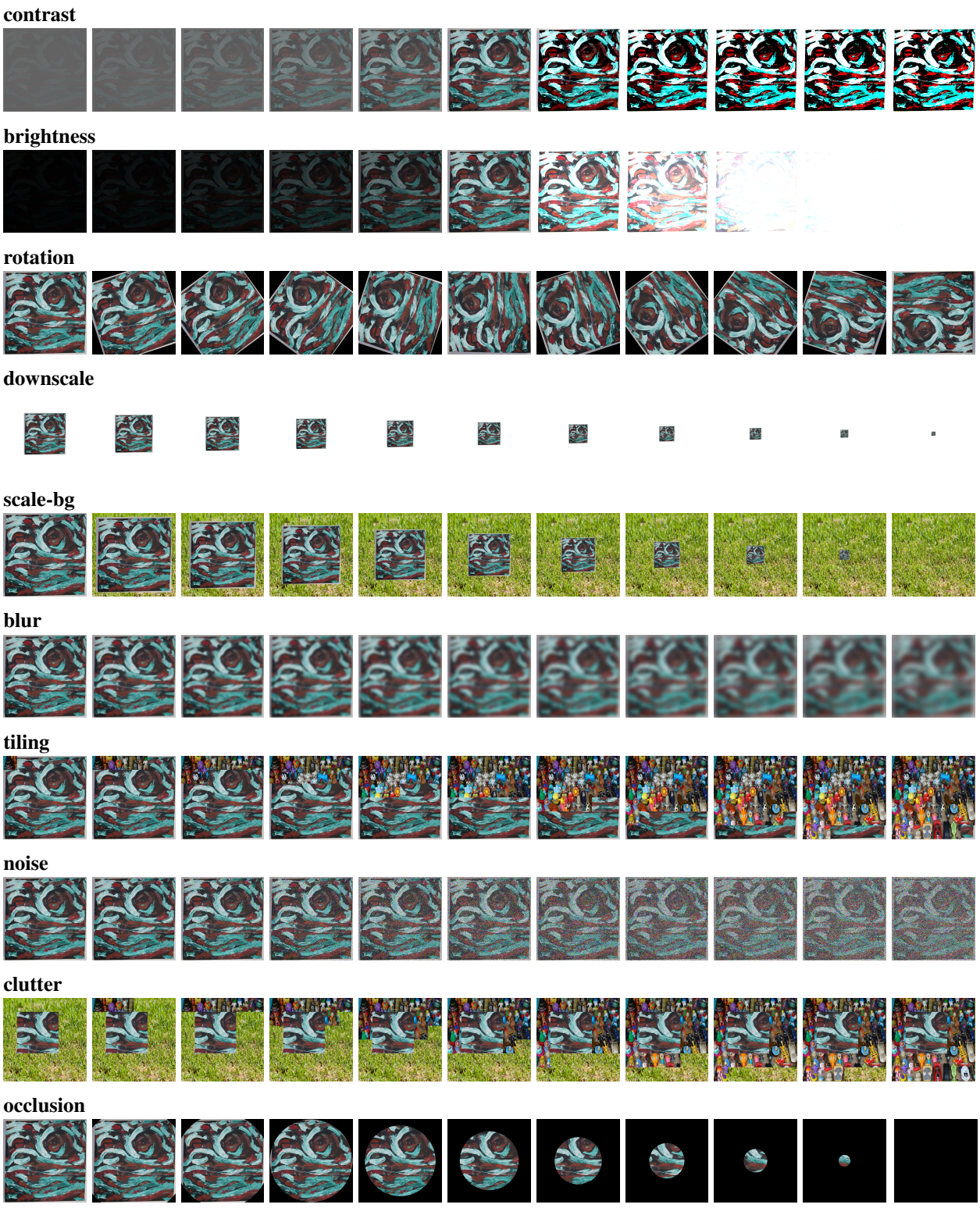

Figure J. **Examples of transformed images for robustness analysis,** including variations in contrast, brightness, rotation, downscaling, background scaling, blur, tiling, noise, clutter, and occlusion. Each row shows the gradual increase in transformation strength from left to right. These transformations are applied to create positive query–target pairs for controlled robustness evaluation.