Supplementary Material: Cost Savings from Automatic Quality Assessment for Generated Images

Xavier Giró-i-Nieto Nefeli Andreou Anqi Liang
Manel Baradad Francesc Moreno-Noguer Aleix Martinez
Amazon

A.1. Code Snippets

We provide a Python implementation of the formula that computes the cost savings to facilitate reproducibility.

```python
def compute_cost_savings(N_MQA, r_AQA_clean,
    P_AQA_clean, P_Gen_clean, c_Gen, c_AQA, c_MQA
    ):

    # First term
    numerator1 = y * P_AQA_clean - P_Gen_clean
    denominator1 = y * P_AQA_clean * P_Gen_clean
    term1 = (numerator1 / denominator1) * c_Gen

    # Second term
    term2 = (1 / (y * P_AQA_clean)) * c_AQA

    # Third term
    numerator3 = P_AQA_clean - P_Gen_clean
    denominator3 = P_AQA_clean * P_Gen_clean
    term3 = (numerator3 / denominator3) * c_MQA

    # Final computation
    result = N_MQA * (term1 - term2 + term3)

    return result
```

Listing 1. Python code to compute the cost savings.

A.2. The effect of the false negatives

In this section, we explore the impact of the false negative in the cost savings. False Negatives (FNs) in our pipeline correspond to those images that, despite achieving the quality standards, are discarded by AutoQA. Most of the cost saving in Eq.5 come from the term weighted the ManualQA cost (c_{MQA}). This term does not contain rP_{AQA} so, in normal scenarios where $c_{MQA} >> c_{Gen}$ and $c_{MQA} >> c_{AQA}$, discarding many *Clean* images will not have much impact. We can however find this condition by forcing $\Delta C = 0$ in Eq.5, and developing the formula as shown in Figure 1.

A.3. AutoGluon Training Details

The AutoGluon model was trained with a diverse dataset of object categories. The histogram of the train and test partitions shown in Figure 2 indicate that the train partition is unbalanced, but the test partition is very balanced. As a result, most of the test samples can be considered as out-of-domain from the perspective of the object category. If we used a test set following the training distribution, we expect higher AutoQA precision, that would translate in even higher cost savings.

By default, AutoGluon uses the a partition 90% of the provided data for training, and the remaining 10% for internal validation. The AutoGluon models were left to train as much time as needed, which was in the order of minutes for each binary classifier. Models were trained with a p3.2xlarge cloud desktop in Amazon Web Services (AWS) equipped with a single Tesla V100-SXM2 GPU with 16 GB of memory.

A.4. VLM Prompts

The VLMs where queried with the question prompts described in Table 1. Some of the defects were divided in more fine-grained categories to obtain more precise descriptions. The VLM was modulated with the role prompt presented in Table 2.

$$
\begin{aligned}
&\text{Let: } P_{Gen}(\checkmark) = p \\
&\quad c_{Gen} = g \\
&\quad c_{AQA} = a \\
&\quad c_{MQA} = m
\end{aligned}
$$

$$
\text{Setting } \Delta C = 0: \quad N_{MQA}\left(\frac{yp-p}{yp}g - \frac{a}{x} + \frac{1-p}{py}m\right) = 0
$$

$$
\text{Since } N_{MQA} \neq 0: \quad \frac{(yp-p)g}{yp} - \frac{a}{y} + \frac{(1-p)m}{py} = 0
$$

$$
\begin{aligned}
&\text{Multiply by } yp: \\
&(yp-p)g - ap - (1-p)m = 0 \\
&ypg - pg - ap - m + pm = 0 \\
&ypg = pg + ap + m - pm \\
&y = \frac{pg + ap + m - pm}{pg} \\
&\therefore y = \frac{P_{Gen}(\checkmark) \cdot c_{Gen} + c_{AQA} \cdot P_{Gen}(\checkmark) + c_{MQA} - c_{MQA} \cdot P_{Gen}(\checkmark)}{P_{Gen}(\checkmark) \cdot c_{Gen}} \\
&\text{where: } y = \frac{TP}{TP + FP + TN + FN}
\end{aligned} \tag{1}
$$

Figure 1. Application of the cost savings formula to find the condition when AutoQA does not bring cost savings related to the false negatives (FN).

A.5. Comparison between VLMs

The choice of Nova Pro 1.0 as the reference VLM in the body of our experiments is based on a preliminary study reported in this section. We considered two VLMs accessed through AWS Bedrock: Nova Lite 1.0, and Nova Pro 1.0. For each type of defect, we build a balanced dataset of close to 100 samples. This means that random predictions correspond to a precision of 0.5 when making *Defect* or *Clean* predictions.

Table 3 shows how none of the VLMs can effectively capture any of the defects, an observation aligned with the limitations of VQA-based approaches observed in [25]. The few high $P(\text{✗})$ are also paired with very low $r(\text{✗})$, which should be close to 0.5 because of the balanced dataset we are using. This means that very few defects were detected, even with high precision. On the other hand, when $r(\text{✗})$ is closer to the expected value of 0.5, the associated $P(\text{✗})$ falls close to 0.5, which indicates a random behaviour in a balanced dataset. When comparing Nova Pro and Lite, the former provides better results for both the amount of detected defects, and average precision. For this reason, we adopted Nova Pro 1.0 in our experimentation.

A.6. Detailed quantitative results

The detailed metrics provided as plots are offered in this section in their detailed numerical form.

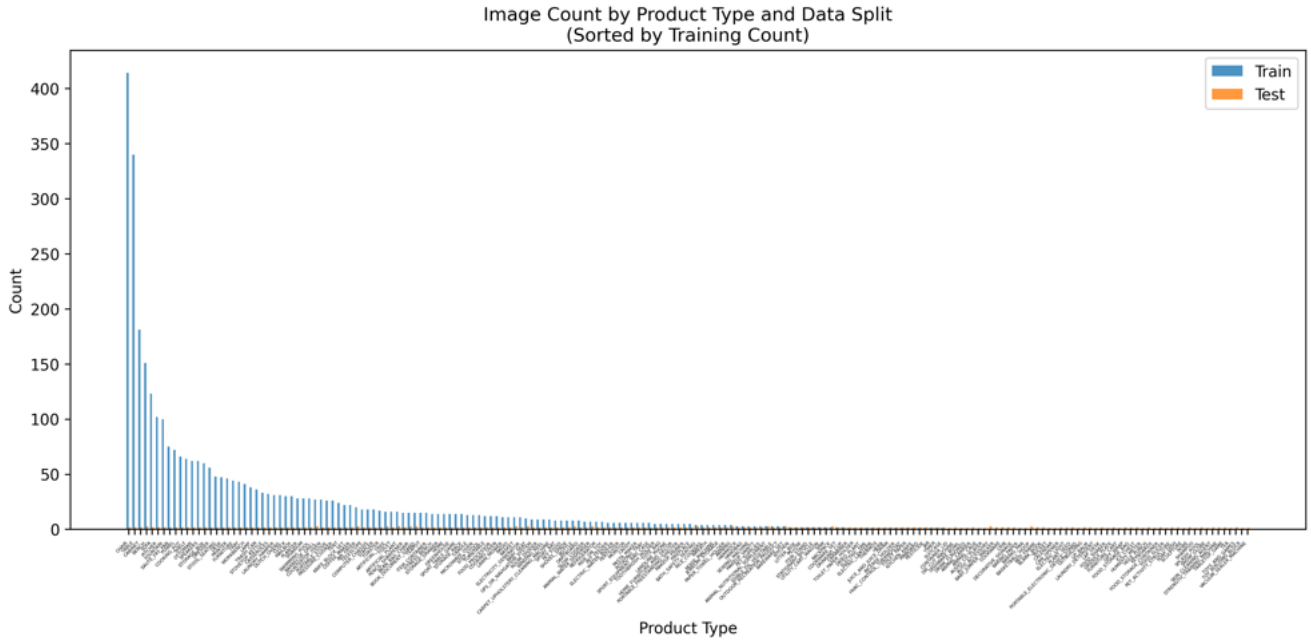


Figure 2. Histogram by object category of the train and test partition used to in the *All Defects* experiment.

A.6.1. Label Distribution in the Annotations

Table 4 provides the detailed distribution of labels collected from the images generated by SDXL.

A.6.2. Volume of images

Table 5 shows the volume of of images at the output of each block of the full pipeline. As previously discussed, the ManualQA yield, $r_{MQA}(\checkmark)$, corresponds to the precision of the AutoQA block for the *Clean* label, $P_{AQA}(\checkmark)$.

A.6.3. Costs

Table 6 presents the detailed costs for our study case of producing 100 high quality images.The first two rows summarize the unitary cost of obtaining a high quality image, which is clearly dominated by the 99.5% of the ManualQA portion. These unitary costs are multiplied by the volume of images estimated in Table 5 to obtain the total cost C_{TOTAL} of each AutoQA configuration. The last column ΔC in Table 6 indicates that the simple binary classifier obtained with AutoGluon offers the highest savings among the considered cases, with a significant reduction of 51.61%.

A.7. Full SDXL Qualitative Results

Figure **??** includes the full predictions in the test set. A visual inspection of the data does not show evidences of biases towards certain objects or backgrounds.

In the early stages of our research, we did observe that the VLM-based AutoQA was biased to only allow images with very simple backgrounds, which were less valuable for our target e-commerce application. For this reason, we extended the prompt to include the detailed defect *Rich background* shown in Table 1. When we experimented with AutoGluon we no longer saw this issue.

A.8. Results for a GenAI model based on Flux

In addition to the use case developed in the main paper, we also provide the results for a smaller scale experiment on outpainted images with a more modern model based on Flux.1[schnell] [27], using ControlNet on top for fine-tuning. Each image was labeled by three different annotators from a pool of 8. The average amount of annotations per worker was 1133.25, with a maximum of 1918 and a minimum of 3. In this use case, we only considered the two configurations based on AutoGluon, as they provided the best performance for SDXL. Table 7 provides the rate and precision of *Clean* that allow computing the volumes of images in Table 8 and cost savings in Table 9. Figure 7 provides a comparison on the average cost to obtain a high quality image for SDXL-vs Flux-based technology. The Flux-based case is much cheaper, mostly because the quality of the generated images is much higher: from a P_{Gen} of 0.187 for SDXL to 0.508 with Flux. As a consequence, the number of images that must go run through ManualQA is much lower: from 526 to only 196 for the *ManualQA only* configuration. This decrease in the workload is the main contributor to cost savings, also for the configurations that include the AutoQA block.

The qualitative results for this model are shown in Fig-

Coarse defect	Detailed defect	Prompt
Main Object Distortion	Surface texture	Focus on the surface of the {object_class}. Is there any distortion on its texture?
	Color blending	Can you see weird color blending at its contours?
	Structural distortion	Is there any structural distortion in the {object_class}?
Main Object Extension	Product extension	Does the {object_class} present a realistic shape? Compare the shape of the {object_class} in the first generated image to the reference image and its segmentation mask. Make sure that the {object_class} did not grow in extension when the background was generated
	Product attached	Is there any other object attached to the {object_class}? If so, is this attachment common and natural?
Misplaced Object	Objects layout	What objects appear in the scene? Are their relative positions natural?
	Floating objects	Look at the {object_class}. It must be standing on a surface. Otherwise, consider that it is floating, which is a severe issue.
	Matching location	In which locations is the normally found? Does the context in the image represent one of these probable locations?
	Functional location	Where is the {object_class} located? Does it appear in a proper functional location?
	Rich background	How is the background around the {object_class}? The background must contain rich semantic and be aesthetically appealing. A solid or uniform background is not acceptable.
Scale Mismatch	Scale mismatch	There is an anomaly in the size of the {object_class} compared to the rest of objects in the scene. True or false?
Background Objects Distortion	Objects distortion	What objects appear in the image? Is there any distortion in any of them?
Background Structural Distortion	General	Is there any structural distortion in the scene?
	Because occlusion	Is the background behind the {object_class} realistic? Make sure that there are no discontinuities in the generated background because of the occlusion of the {product_type}

Table 1. Question prompts organised in a hierarchy of coarse and detailed defects.

ure 8.

Knowledge	You are a vision-language assistant responsible for assessing the quality of synthetically generated images. You have expertise in professional photography for e-commerce and design. You will receive a question and your task is to answer with the most appropriate score.
Objective	You are assessing the quality of a synthetically generated image depicting a {product_type}. This image is generated by adding a background to an image of a {product_type}. The main {product_type} is the primary object of the image. The background is generated by a text-to-image model.

Table 2. Text prompts for System knowledge and objective.

Defect (✗) type (# img)	**Nova Pro 1.0**		**Nova Lite 1.0**	
	y_{AQA}	$P_{\text{AQA}}\uparrow$	y_{AQA}	$P_{\text{AQA}}\uparrow$
Main Object Distortion (92)	0.97	1.000	0.94	0.500
Main Object Extension (95)	0.81	0.667	0.91	0.444
Product placement (96)	0.84	0.867	0.74	0.600
Scale Mismatch (88)	0.27	0.516	1.00	0.000
Bg. Objects distortion (94)	0.52	0.478	1.00	0.000
Bg. Structural Distortion (95)	0.26	0.543	0.52	0.522

Table 3. AutoQA yield, y_{AQA}, and precision for defects, P_{AQA}(✗), for Nova Pro and Lite 1.0. Values in red highlight the cases where the VLM fails which, for the case of the balanced dataset, corresponds to y_{AQA}(✗) $\ll 0.5$ and/or P_{AQA}(✗) ≈ 0.5.

Coarse defect	**No issue (1)**	**Some issue (2)**	**Significant issue (3)**
internal distortion	83.60%	10.15%	6.25%
objects distortion	35.37%	16.64%	47.99%
product extension	65.98%	9.49%	24.53%
product placement	78.26%	6.81%	14.93%
scale mismatch	93.07%	2.15%	4.78%
structure distortion	37.39%	17.24%	45.37%

Table 4. Distribution of annotated labels across the types of coarse defects on images generated with SDXL. Each image can be present multiple defects. Annotators may not agree.

	GenAI	**AutoQA**		**ManualQA**	
	N_{Gen}	r_{AQA}	N_{AQA}	r_{MQA}	N_{MQA}
ManualQA only	526	-	-	0.187	100
Cascade (AG only)	1389	0.239	333	0.297	100
Cascade (AG & NP)	2712	0.153	415	0.241	100
Single (AG only)	2119	0.118	250	0.400	100

Table 5. Number of images at the output of each block of the pipeline to obtain 100 high quality images generated with SDXL.

	c_{Gen}	C_{AQA}	C_{MQA}	C_{TOTAL}		
Processing cost / image	0.00209	0.00024	0.50	0.50233		
Processing cost / image (%)	0.45%	0.05%	99.50%	-		
ManualQA only	0.0112	0	2.6738	2.6850	ΔC	$\Delta C(\%)$
Cascade (AG only)	0.0294	0	1.6835	1.7163	0.9686	36.08
Cascade (AG & NP)	0.0567	0.0065	2.0747	2.1379	0.5471	20.38
Single (AG only)	0.0443	0	1.2500	1.2994	1.3856	**51.61**

Table 6. Average cost and savings for producing one high quality image with SDXL, provided in $.

	Clean(✓)	
Configuration	**r**	**P ↑**
Random (0.5)	0.500	0.418
Cascade	0.758	0.596
Single	0.500	0.694
Oracle	0.508	1.000

Table 7. AutoQA performance metrics on 128 test samples for two configurations based on Flux as image generator and AutoGluon as AutoQA.

	GenAI	**AutoQA**		**ManualQA**	
	N_{Gen}	r_{AQA}	N_{AQA}	r_{MQA}	N_{MQA}
ManualQA only	196	-	-	0.508	100
Cascade	221	0.758	168	0.596	100
Single	288	0.500	144	0.694	100

Table 8. Number of images at the output of each block of the pipeline to obtain 100 high quality images generated with Flux.

	$C_{Gen} \downarrow$	$C_{AQA} \downarrow$	$C_{MQA} \downarrow$	$C_{TOTAL} \downarrow$		
Processing cost / image	0.00209	0.00024	0.50	0.50233		
Processing cost / image (%)	0.45%	0.05%	99.50%	-		
ManualQA only	0.0041	0	0.9804	0.9845	ΔC	$\Delta C(\%)$
Cascade (AG only)	0.0046	0	0.8389	0.8436	0.1409	14.32
Single (AG only)	0.0060	0	0.7205	0.7265	0.2580	**26.21**

Table 9. Average cost and savings for producing one high quality images by Flux, provided in $.

Figure 3. Full set of True Defect (✗) predictions.

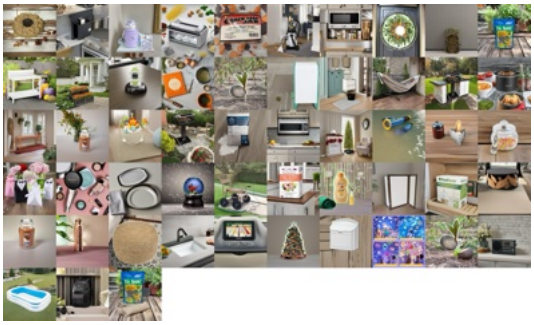


Figure 4. Full set of False Defect (✗) predictions.

Figure 5. Full set of True Clean (✓) predictions.

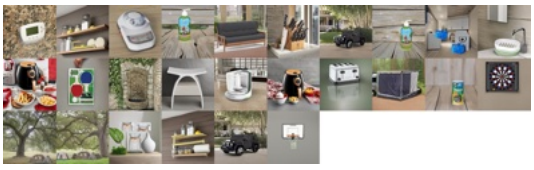


Figure 6. Full set of False Clean (✓) predictions.

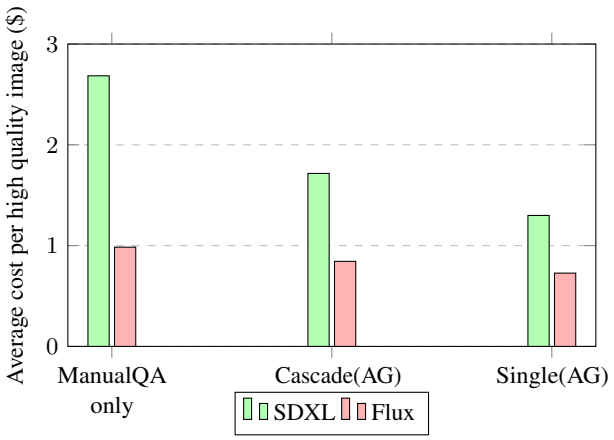


Figure 7. Average cost comparison between SDXL- vs Flux-based outpainting technology.

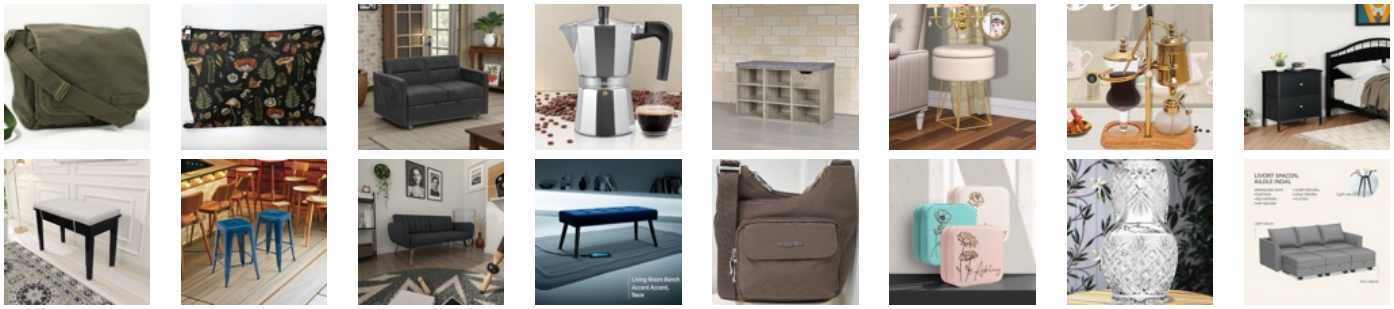
References

[25] Candace Ross, Melissa Hall, Adriana Romero-Soriano, and Adina Williams. What makes a good metric? evaluating automatic metrics for text-to-image consistency. In *First Conference on Language Modeling*, 2024. 2

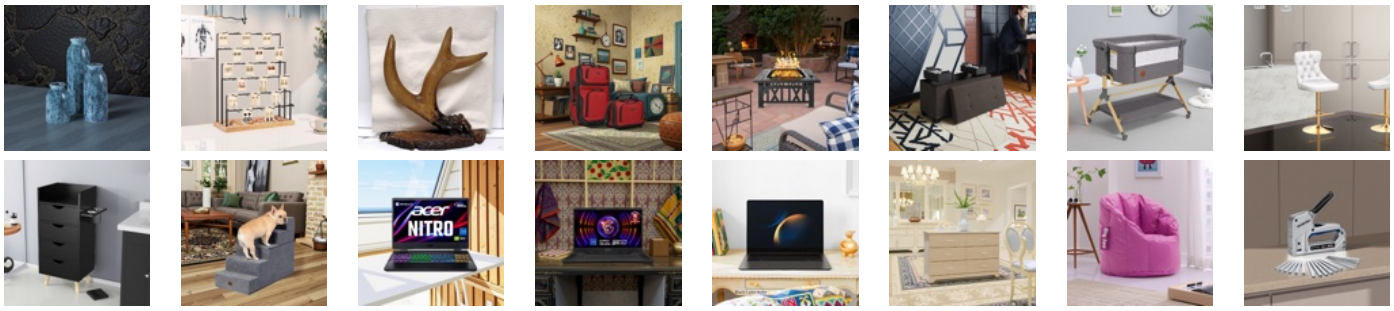
[27] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 3

(a) True Clean (✓) predictions.

(b) True Defect (✗) predictions.



(c) False Clean (✓) predictions.



(d) False Defect (✗) predictions.

Figure 8. Random sample of AutoQA predictions provided by the Single (AG only) configuration on images generated with a Flux model.