Beyond the Brush++: A Flexible Pipeline for Fully-automated Generation of Realistic Inpainted Images

Giulia Bertazzini[1*], Chiara Albisani[1], Daniele Baracchi[1], Dasara Shullani[1], Alessandro Piva[1]

[1]Department of Information Engineering, University of Florence, Florence, Italy.

*Corresponding author(s). E-mail(s): giulia.bertazzini@unifi.it;
Contributing authors: chiara.albisani@unifi.it; daniele.baracchi@unifi.it; dasara.shullani@unifi.it; alessandro.piva@unifi.it;

Abstract

Partially manipulated images pose a growing threat to the reliability of online content. The rapid spread of diffusion-based inpainting tools has made the creation of such manipulations increasingly easy to perform. As a result, the multimedia forensics community is disadvantaged compared to the attackers, as developing effective localization techniques often requires the creation of large datasets, a resource-intensive process due to the necessary human effort. In this paper, we present **Beyond the Brush++** (BtB++), a fully automated pipeline for generating large-scale datasets of realistic inpainted images. Our experiments demonstrate that BtB++ is both flexible and easily integrates different models and configurations, offering the adaptability required to address evolving models and application scenarios. Moreover, an automatic filtering mechanism ensures quality control by discarding low-quality generated images. To provide an initial assessment of the proposed filtering strategy, we also conducted a small-scale human evaluation, studying the alignment between human perceptual judgments and the automatic metrics used for filtering.

Keywords: diffusion models; multimedia forensics; inpainting localization; quality assessment

1 Introduction

The spread of manipulated visual media poses a major societal threat, as it has become exceptionally challenging for people to distinguish between real and fake content [1]. Modern advancements in image editing and the wide availability of intuitive software have simplified the creation of convincingly realistic fakes. In this context, techniques like image inpainting [2] have re-emerged in prominence. Originally created for restoration tasks like removing or inserting objects, these methods are now increasingly co-opted for malicious purposes [3]. They are used to introduce new elements or completely change the meaning of an image by erasing key subjects, a trend fueled by generative models that produce high-quality forgeries [4].

Nowadays, image inpainting techniques are classified in three major categories: GAN-based [5], Diffusion-based [6], and Transformers-based [7] algorithms.

Developing effective automated detectors for fake visual content depends on the availability of large-scale training datasets containing manipulated images. A major bottleneck in this area is the scarcity of public datasets for forgeries created with image inpainting. The most widely used resources, DEFACTO [8] and IMD2020 [9], offer some examples. DEFACTO utilizes an example-based method for object removal, while IMD2020 applies a mix of three approaches, one of which is the built-in OpenCV inpainting function.

A central challenge in generating large-scale datasets of tampered images lies in the need for an automated inpainting procedure. An initial step toward this goal was proposed by Wang et al. [10], who introduced ImagenEditor, a system based on diffusion models fine-tuned for text-guided image inpainting. However, this approach still requires user-provided masks and prompts. More recently, BtB [11] and SAGI [12] have introduced inventive pipelines for fully automatic text-guided inpainting. Notably, BtB relies exclusively on open-source models for all its components, whereas SAGI employs commercial solutions for both prompt generation and the employed vision–language model for realism assessment.

This work extends [11] and introduces *Beyond the Brush++* (BtB++), a novel, fully automated framework designed to generate extensive datasets of high-fidelity inpainted images. The pipeline is composed of four sequential modules. The initial module performs segmentation to derive three sets of free-form masks (ranging from small to large) that correspond to semantically significant regions targeted for manipulation. Subsequently, a visual language model is employed to produce five textual prompts for each region, simulating a malicious user's intent to alter the content while maintaining the overall semantic integrity of the image. The third module utilizes a diffusion-based model to execute the inpainting according to the generated prompts and masks. Finally, the last module employs an automated filtering system that leverages CLIPScore [13] and Multi-dimensional Preference Scoring (MPS) [14] to assess and regenerate, if needed, low-quality outputs. Overall, the architecture is inherently flexible and extensible, supporting various models to suit different needs.

Our results prove that BtB++ is a versatile and extensible pipeline, supporting a wide range of models and configurations. This flexibility makes it particularly suited to adapt it to the rapid evolution of models, allowing researchers to incorporate the

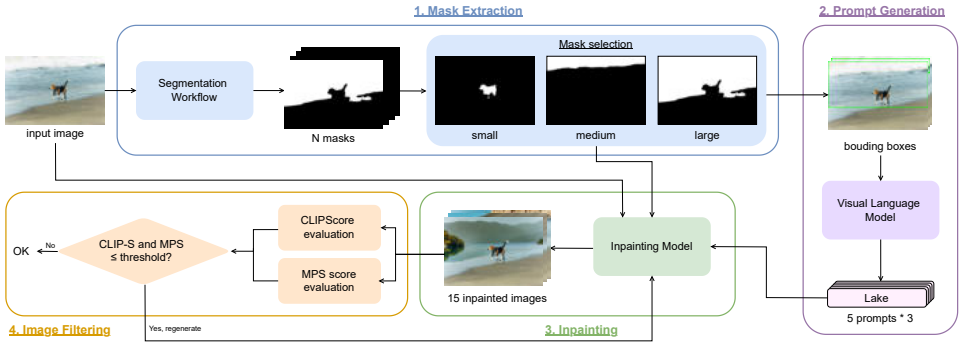

Fig. 1: Overview of BtB++ pipeline. Starting from a real input image, the framework generates 15 different inpainted images. Finally, CLIPScore and MPS are computed for each output to filter out low-quality results.

most recent architectures and configurations, without the need to redesign the pipeline or alter its underlying principles.

With respect to BtB [11], there are several improvements: for mask extraction, we introduced an all-in-one approach generating masks directly from the input image; we adopted four different Visual Language Models for prompt generation; we used four different Diffusion Models for inpainting; we introduced the automatic filtering strategy at the end of the generation process to discard low-quality images and, when necessary, regenerate them to improve overall dataset quality.

The remainder of this paper is organized as follows. In Section 2, we introduce the BtB inpainting pipeline. Section 3 describes the modules evaluated in our study and presents the corresponding results. Section 4 discuss the key factors influencing each component's performance and how they relate. Finally, Section 5 summarizes our findings and provides concluding remarks.

2 Proposed Method

In this section, we describe the proposed method (Beyond the Brush++ framework) to fully automate the generation of large datasets containing realistic inpainted images. As shown in Figure 1, our approach consists of four main modules: mask extraction, prompt generation, inpainting, and image filtering.

2.1 Mask Extraction

The initial step of our approach involves extracting meaningful regions from an input image to be inpainted. To this end, we employed a segmentation workflow that, given an input image, generates various masks.

The resulting masks can be then categorized by their dimension to cover the range of editing sizes that users commonly apply to modify images. Specifically we considered small, medium, and large masks. The small category includes masks covering up to

15% of the image area, the medium category encompasses masks covering 15 to 30%, and the large category consists of masks covering 30 to 60% of the image.

Since the selected masks may contain noise that could compromise the quality of inpainting results, we applied to the masks two sequential morphological operations, an opening followed by a closing, to refine them.

2.2 Prompt Generation

The second step of our method involves generating prompts to inpaint the input image within the masked regions. This is achieved by processing the image with a Visual Language Model (VLM), where each identified region is first highlighted with a green bounding box and then input into the VLM. The model was queried on how it would replace the object within the green box while maintaining the image's semantic coherence. Specifically, we queried the VLM to explain the chain of thought for the rationale used for the proposal, and the replacement in at most four words, following the template: '`Rationale: xxx. Replace with: yyy.`'. We based our request on the findings presented by Wei et al. [15], which demonstrate that generating a chain of thought can improve the performance of large language models in complex reasoning tasks. This approach allows the VLM to autonomously determine the replacement content for the area within the bounding box, thereby eliminating any kind of human intervention.

2.3 Inpainting

The final step of the BtB++ approach focuses on generating inpainted images by combining the extracted masks with the prompts suggested by the VLM. For this purpose, we employed diffusion-based inpainting models, which modify the masked regions of the original image while preserving the surrounding context, according to the given prompt.

Figure 2 shows some examples of the resulting inpainted images using the described method.

2.4 Image Filtering Procedure

Despite the fact that diffusion-based inpainting tools generally perform well in image manipulation, the generated results do not always perfectly align with the intended prompt and may sometimes display inconsistencies between the edited and unedited regions. Typical issues include incomplete adherence to the textual description, as well as visible artifacts in the manipulated region or at the boundaries between edited and original areas. To mitigate these limitations, we introduced an image-filtering procedure based on CLIPScore [13] and Multi-dimensional Preference Scoring (MPS) [14]. These metrics are employed not as indicators of realism, but as complementary measures of prompt alignment and overall perceptual quality of the inpainted area, enabling us to automatically prioritize outputs that better fulfill the intended editing task.

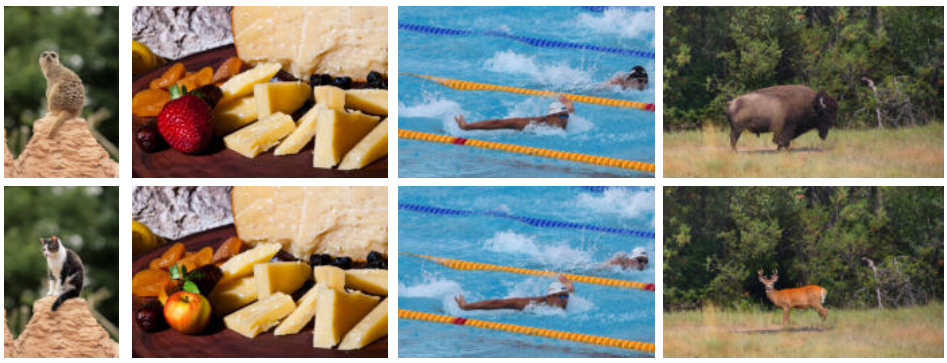

Fig. 2: Examples of images inpainted using BtB++ framework. The first row shows original images from the Open Images v7 [16] dataset, while the second row shows the corresponding inpainted versions, obtained by using our approach.

CLIPScore

CLIPScore [13] measures image-text alignment by computing cosine similarity between CLIP [17] embeddings. For a text CLIP embedding t and an image with visual CLIP embedding v, CLIPScore is defined as

$$\text{CLIP-S}(t, v) = w * max(cos(t, v), 0)$$

where $w = 2.5$. Higher values of this score indicate a stronger semantic consistency between the text and the image. Building on these findings, we leveraged CLIPScore to filter out inpainted images that not faithfully reflect the intended prompt for the manipulation. Specifically, we computed CLIPScore between the manipulated region of the image and the corresponding prompt used for the inpainting. Instead of simply cropping the manipulated region, we computed the visual embedding from a version of the image where all unedited pixels were replaced with black, ensuring that only the masked area contributed to the similarity measure. This approach prevents contributions from unedited regions when the mask spans non-contiguous parts of the the image, as show in Figure 3.

This procedure yields a single score per image, enabling us to discard results with low alignment and retain only those with a CLIPScore above a chosen threshold.

MPS Score

Multi-dimensional Preference Scoring [14] (MPS) is a CLIP-based multi-dimensional preference scoring model for the evaluation of text-to-image models that provides a fine-grained assessment of generated images. It models human preferences across four distinct dimensions: aesthetics, semantic alignment, detail quality, and overall impression, with scores ranging from negatives values (poor quality) to positive values (high quality). Given a prompt x, a generated image y and a preference condition c, MPS is defined as

$$\text{MPS}(x, y|c) = \alpha f_{v,t} \cdot f_t$$

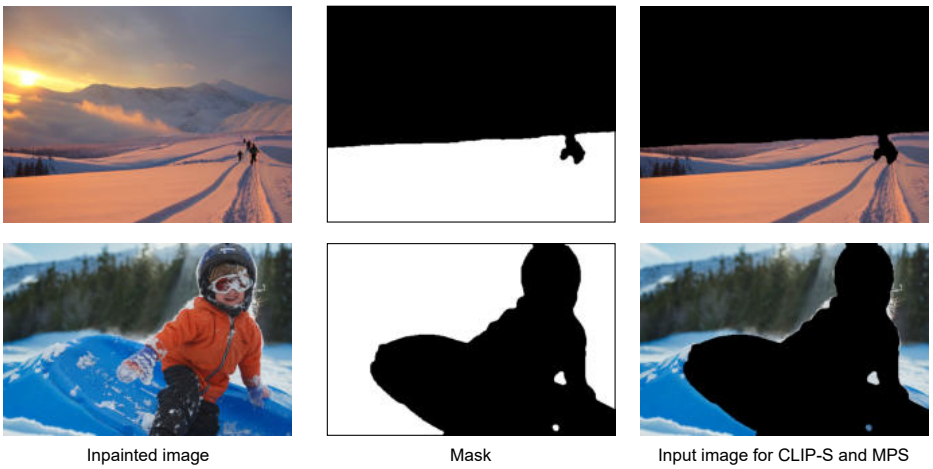

Fig. 3: Examples of input images used to compute CLIPScore and MPS, where all unedited pixels are replaced with black to isolate the manipulated region.

where $f_{v,t}$ is the first dimension (i.e. the cls token) of the cross-attention between the image features of y and the text features of x, conditioned on the features of c, f_t is the last dimension of the prompt feature embedding and α is a learned scalar. MPS is specifically designed for assessing full generated images. In our study, however, we computed it only on the inpainted region, following the same protocol adopted for CLIP-S, where non-manipulated areas are set to black to ensure that only the manipulated pixels contribute to the score. Similarly to CLIP-S, MPS outputs a single score per image.

3 BtB++ Modules Evaluation

In this section, we present a detailed analysis of each component of the proposed pipeline. For every stage, we compare alternative models and evaluate their performance with respect to attributes specifically relevant to that component. All evaluations were conducted starting from the source images of the BtB dataset [11], considering three subsets of 100 real source images each, respectively selected from the Flickr30k [18], VISION [19], and FloreView [20] dataset partitions used to construct *BtB-Flickr30k*, *BtB-VISION*, and *BtB-FloreView*.

The images in these subsets retain the original characteristics of their source datasets: *Flickr30k* images range from 180×240 up to 500×500 pixels, mostly depicting people in everyday scenes; *VISION* images are natural images from smartphones with varied resolutions, generally between 960×720 and 5312×2988 pixels; *FloreView* images are high-resolution photographs (up to 8000×6000 pixels) of 35 subjects in the city center of Florence. Both portrait and landscape orientations are included. The subsets were sampled randomly to ensure coverage of diverse visual content and different resolution ranges, providing a representative selection for evaluation.

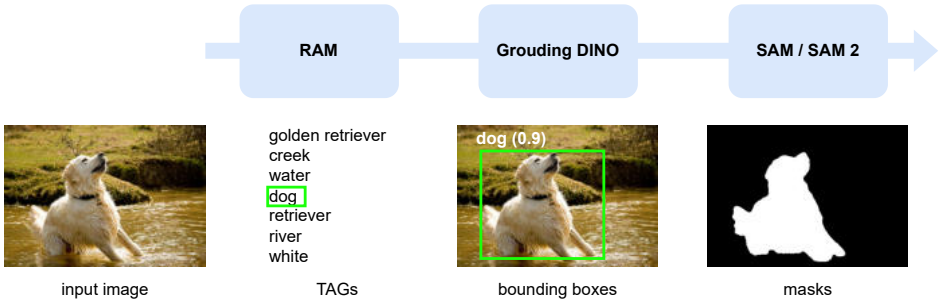

Fig. 4: Mask extraction module when sequentially using three foundation models: Recognize Anything Model, Grounding DINO, and Segment Anything (or Segment Anything 2) Model

All subsequent evaluations focus on these subsets, with the exception of image filtering evaluation described in Section 3.4, that has been performed on the entire BtB dataset. As an additional experiment, we examine a scenario in which the mask extraction module is disabled while the prompt generation and inpainting modules remain active, demonstrating the framework's flexibility and its ability to accommodate different configurations. Finally, a discussion on the image filtering module is proposed.

3.1 Mask Extraction

To evaluate the segmentation workflow, we considered two approaches: (i) the use of a segmentation model in combination with other foundation models and (ii) all-in-one approaches that generate masks directly from an input image, thereby avoiding the need for chaining several models.

The first strategy is illustrated in Figure 4. In this context, we leveraged **Segment Anything Model** [21] (SAM) and its successor, **Segment Anything Model 2** [22] (SAM 2), both preceded by Recognize Anything Model (RAM) [23] and Grounding DINO [24]. Initially, the input image is processed by the RAM and Grounding DINO (RD), to identify a wide range of objects within an image and their corresponding bounding boxes. Then, SAM or SAM 2 generates segmentation masks for each detected object using these bounding boxes as prompts. SAM 2 improves its predecessor with a simple transformer architecture, achieving higher accuracy and inference speeds up to six times faster.

Alternatively, all-in-one approaches simplify the workflow by producing segmentation masks directly from an image, avoiding the complexity of chaining multiple models. To this end, we considered **Florence-2** [25] and **SEEM** [26]. Florence-2 performs tasks such as captioning, detection, and segmentation from simple text prompts. In our BtB++ pipeline, we leveraged its *dense region captioning* to describe detected regions and its *referring expression segmentation* to generate the corresponding masks, without relying on any additional foundation model. We also exploited SEEM, an all-in-one promptable and interactive model capable of segmenting everything in an image

using points, boxes, scribbles, or masks. SEEM supports open-set semantic, instance, and panoptic segmentation, offering semantic-aware outputs and greater flexibility than SAM, which lacks semantic labeling.

In order to compare the different segmentation workflows examined, we evaluate three different parameters: (i) the average computational time[1] required for generating masks, (ii) the number of connected components per mask, and (iii) the distribution of masks across the three size categories (small, medium, and large). The results of this comparison are summarized in Table 1. The analyzed models exhibit substantial differences in the average time required for producing masks for a single image. Across all three source datasets (Flickr30k, VISION, and FloreView), SEEM consistently emerges as the fastest model, requiring between 0.19 and 0.37 seconds per image. In contrast, Florence-2 and the sequential chaining of multiple models exhibit substantially higher computational times, with the gap becoming particularly pronounced on VISION and FloreView, where Florence-2 reaches up to 39.90 and 86.7 seconds per image, respectively. This behavior is coherent with the dataset characteristics, since higher-resolution images (e.g., in FloreView) inherently increase computational cost.

In terms of connected components, Florence-2 consistently produces masks with a single connected component in the vast majority of cases (around 93–96% across datasets). SEEM also generates predominantly single-component masks, although with a non-negligible proportion of empty masks, especially on VISION (43%). Conversely, the segmentation workflow that uses SAM or SAM-2 tends to yield a higher fraction of fragmented masks (i.e., more than one connected component), particularly on Flickr30k and VISION, confirming a systematic tendency toward mask fragmentation when chaining multiple foundation models. This aspect is particularly relevant for inpainting applications, as diffusion-based generative models typically perform generally perform better on single, contiguous regions than on fragmented or disjoint areas.

Finally, with respect to mask size distribution, both SEEM and the sequential chaining of multiple models generally yield a more balanced distribution across small, medium, and large masks. Although small masks tend to be predominant, the proportions remain relatively well distributed and vary depending on the specific characteristics of the source dataset. In contrast, Florence-2 consistently produces predominantly small masks (above 90% in all datasets), with negligible or null percentages of large masks, thus limiting its ability to cover broader regions when required.

Leveraging Pre-Existing Masks

Computing segmentation masks from images is often computationally demanding. When a dataset already provides masks, an efficient alternative is to leverage these annotations rather than generating masks from scratch. The drawback of this approach is the lack of control over mask size, since they are already provided; however, it still allows us to leverage the prompt generation and inpainting component of our pipeline. As a representative example, we used **Open Images v7 dataset** [16], which is a very

[1] Computational times have been performed on a dedicated workstation with an AMD Ryzen 9 7950X processor, 128 GB of RAM, and an NVIDIA GeForce RTX 4090.

Table 1: Evaluation of the mask extraction component. Reported metrics include the computational time required per image (in seconds), the percentage of masks with zero, one, or multiple connected components, as well as the percentage of small, medium, and large masks. Computational time is averaged over 100 images for each source dataset. All other metrics are computed over all masks produced by the model. RD denotes the sequential pipeline RAM + Grounding DINO images.

	Segmentation workflow	Time per image (s)	Connected components			Mask sizes		
			0	1	>1	small	medium	large
Flickr30k	RD-SAM	4.80	0.00	0.55	0.45	0.66	0.21	0.13
	RD-SAM 2	2.87	0.26	0.55	0.18	0.66	0.21	0.13
	Florence-2	5.50	0.00	0.96	0.04	0.91	0.08	0.00
	SEEM	0.19	0.16	0.78	0.06	0.46	0.26	0.16
VISION	RD-SAM	12.06	0.01	0.56	0.44	0.36	0.32	0.32
	RD-SAM 2	10.70	0.02	0.53	0.44	0.48	0.28	0.24
	Florence-2	39.90	0.00	0.95	0.05	0.97	0.02	0.00
	SEEM	0.29	0.43	0.44	0.13	0.69	0.17	0.15
FloreView	RD-SAM	28.78	0.00	0.75	0.25	0.46	0.28	0.25
	RD-SAM 2	29.87	0.00	0.61	0.38	0.50	0.25	0.25
	Florence-2	86.7	0.00	0.93	0.07	0.99	0.01	0.00
	SEEM	0.37	0.20	0.68	0.11	0.51	0.23	0.27

recent dataset containing approximately 9 million of real images annotated with image-level labels, object bounding boxes, instance segmentation masks, visual relationships, and localized narratives. In particular, it provides around 2.8 million object instances spanning 350 categories. In our experiments, we sampled 100 images from this dataset and we applied the BtB pipeline [11] just skipping the mask extraction step. Figure 5 shows some examples of the results obtained on this dataset. We refer to this dataset as **OpenImages-100**.

3.2 Prompt Generation

For a more comprehensive analysis, we evaluated multiple VLMs using the same input prompt. For each image, we generated five distinct outputs per VLM to mitigate occasional nonsensical or inconsistent responses. We compared four distinct VLMs of different scales to evaluate their performance and computational cost. **InternVL3-8B** [27] is an advanced multimodal large language model that preserves the architecture of InternVL2.5 [28] and earlier versions while reaching state-of-the-art performance, with enhanced multimodal perception and reasoning capabilities. **Mini InternVL Chat 4B V1-5** [29, 30] is lightweight member of the InternVL family, designed to reduce computational cost while maintaining high-quality multimodal understanding and reasoning capabilities. We also included **Qwen2.5-VL-3B-Instruct** [31] from the Qwen vision–language series that outperform comparable competitors, reaching strong capabilities even in resource-constrained environments. Finally, **SmolVLM** [32] is a small,

Fig. 5: Examples of inpainted images sourced from Open Images v7 dataset. The top row shows the original images, while the bottom row represents the inpainted versions obtained through the prompt generation and inpainting component of our pipeline.

fast, memory-efficient, and fully open-source model that supports commercial use and can be deployed to smaller local setups, with completely open training pipelines.

Table 2 shows the evaluation of the selected VLMs in terms of: (i) number of parameters, (ii) average computational time[1] required to generate a single prompt, (iii) number of answers that not follow the required template (omitting '`Replace with:`' in the output), and (iv) number of too long answers (i.e., cases when the suggested replacement exceeded the four-word limit specified in the query).

Typically, larger models require more time to generate responses. However, exceptions are observed depending on the dataset characteristics. For instance, Mini InternVL Chat 4B V1-5, despite not being the largest model, is among the slowest across datasets, requiring 4.33 seconds per prompt on Flickr30k, but only 1.66 and 2.08 on VISION and FloreView, respectively. On the contrary, SmolVLM consistently achieves the fastest runtime, ranging from 0.20 seconds on VISION to 0.60 seconds on Flickr30k, although it often fails to follow the required output format, with 42–93% of responses being incorrectly formatted depending on the dataset. Qwen2.5-VL-3B-Instruct provides a moderate trade-off, with runtimes between 1.95 and 3.40 seconds, and very few format errors. Overall, InternVL3-8B remains the most reliable model in terms of balancing computational cost and output quality, showing 0% of bad format answers and negligible proportions of overly long responses across all datasets. Figure 6 compares the outputs produced by the evaluated models for a common input image.

3.3 Inpainting

The final step of the BtB++ approach generates inpainted images using the extracted masks and generated prompts. For a deeper evaluation, we applied several diffusion models for inpainting to the same input images, masks, and prompts for direct comparison. We considered the open-source software **Fooocus** [33], which builds on Stable

Table 2: Evaluation of visual language models on 1500 prompts. The table reports the model size (number of parameters) along with the evaluation metrics: the average computational time per prompt (in seconds), the percentage of bad format answers (i.e., that not follow the indicated template, omitting '`Replace with:`' field) and the percentage of too long answers (i.e., replacements exceeding four words)

	VLM	Number of parameters	Time per prompt (s)	% Bad format answers	% Too long answers
Flickr30k	InternVL3-8B	8B	3.73	0.00	0.02
	Mini InternVL Chat 4B V1-5	4B	4.33	0.02	0.15
	Qwen2.5-VL-3B-Instruct	3B	1.95	0.00	0.45
	SmolVLM	2B	0.60	0.93	0.00
VISION	InternVL3-8B	8B	1.36	0.00	0.01
	Mini InternVL Chat 4B V1-5	4B	1.66	0.02	0.21
	Qwen2.5-VL-3B-Instruct	3B	2.36	0.00	0.31
	SmolVLM	2B	0.20	0.77	0.00
FloreView	InternVL3-8B	8B	1.36	0.00	0.02
	Mini InternVL Chat 4B V1-5	4B	2.08	0.01	0.26
	Qwen2.5-VL-3B-Instruct	3B	3.40	0.00	0.38
	SmolVLM	2B	0.30	0.42	0.04

Diffusion and Midjourney designs. A key advantage of Fooocus is its ability to operate offline, thus eliminating the need for manual parameter settings and allowing users to focus on creating prompts and selecting images. Furthermore, by leveraging the Fooocus API [34], we automated the inpainting process, reducing human intervention. We also tested **Stable Diffusion v2 Inpainting** [35], a model fine-tuned from the Stable Diffusion v2 base with an additional 200k training steps. It follows the mask-generation strategy introduced in LaMa [36], combining these masks with latent VAE representations of the masked image as extra conditioning. Additionally, we included **BrushNet** [37], a plug-and-play dual-branch architecture that injects pixel-level masked-image features into any pre-trained diffusion model to achieve more coherent, high-quality inpainting. Its core innovation lies in separating masked-image features and noisy latents into distinct processing branches. Finally, we considered **DreamShaper v8**[2], which is fine-tuned from Stable Diffusion 1.5 [35] and is capable of generating portraits of real people and anime-style content.

To evaluate the quality of the different inpainting methods, we employ both CLIP-Score and MPS metrics on the inpainted regions, as described in Section 2.4. The average scores obtained for each method are summarized in Table 3. Overall, all methods achieve comparable CLIP-S values, whereas the MPS scores show more pronounced differences, with BrushNet achieving the highest average score across all subsets. In addition to these quantitative results, qualitative comparisons are presented in Figure 7. The first example (top row), generated with the prompt "*Chinese historical structure*", reflects the average performance trends observed in our evaluation.

[2]https://civitai.com/models/4384/dreamshaper

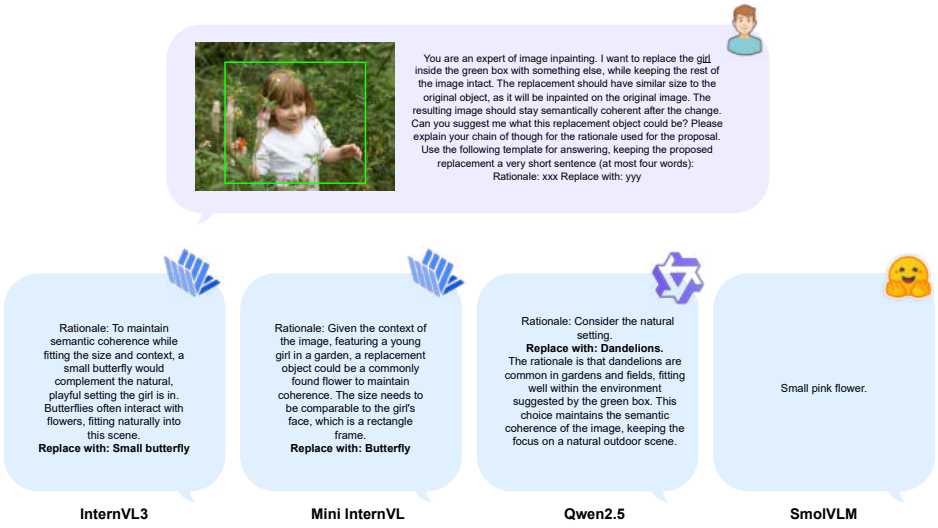

Fig. 6: Output examples from the evaluated VLMs for the same input image. In this example, InternVL3 and Mini InternVL strictly follow the indicated template, Qwen2.5 produces an answer that only partially adheres to it, while SmolVLM neither follows the template nor provides a clear rationale.

Table 3: Average CLIP-S and MPS scores for different inpainted methods on the BtB-Flickr30k, BtB-VISION, BtB-FloreView subsets.

		Fooocus	SDv2	BrushNet	DreamShaper v8
Flick.	Avg CLIP-S	0.57	0.59	0.61	0.61
	Avg MPS	0.83	0.20	1.07	0.54
VIS.	Avg CLIP-S	0.61	0.60	0.61	0.60
	Avg MPS	1.65	0.87	1.69	0.89
Flore.	Avg CLIP-S	0.59	0.58	0.60	0.58
	Avg MPS	0.51	-0.11	0.80	0.00

Conversely, the second example (bottom row), corresponding to the prompt "*Friendly dog*", highlights a better result for the Fooocus method. These results suggest that the relative performance of inpainting models can be highly image-dependent and strongly influenced by mask size, location, and specific prompt used. Additionally, we remark that metrics do not capture the overall perceived quality of the resulting image, as their evaluation is restricted to the inpainted region.

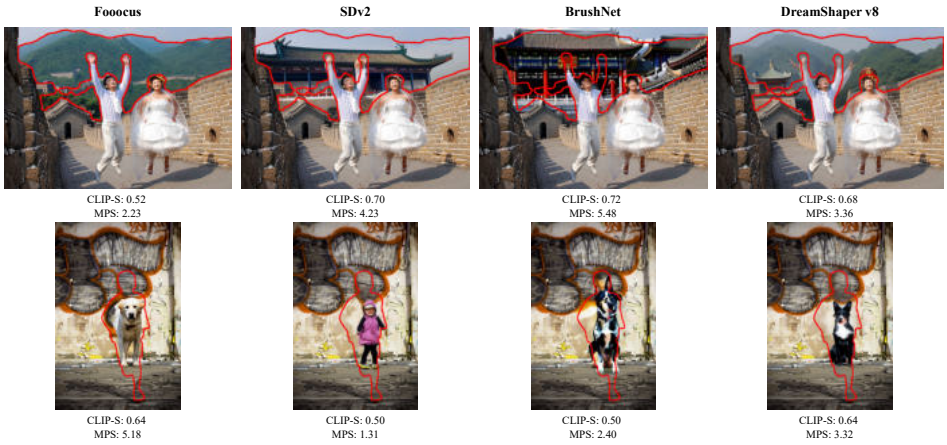

Fig. 7: Example of images generated by using different inpainting tools given the same source image, prompt, and mask. The first row corresponds to the prompt "*Chinese historical structure*", while the second row corresponds to the prompt "*Friendly dog*".

3.4 Image Filtering

To evaluate the effectiveness of CLIPScore and MPS in filtering low-quality images, we computed both metrics on the entire BtB dataset and analyzed their relationship. Despite a moderate positive correlation (SROCC = 0.33, KROCC = 0.22; p-value < 0.5 for both indices), the results indicate that the metrics capture complementary aspects of image quality. To further illustrate this behavior, based on the observed score distributions (Figures 8 and 9), we identify four quadrants using the 70th percentile of each metric as an empirically determined threshold. We emphasize that this threshold has been chosen experimentally but can be freely adjusted by users, depending on how strict they wish to be regarding the quality requirements of the generated images.

Figure 10 illustrates the identified quadrants, along with representative example images for each of them. Images in the bottom-left quadrant (low CLIP-S, low MPS) exhibit both poor quality in the edited region and weak semantic alignment with the requested prompt. Conversely, images in the top-right quadrant (high CLIP-S, high MPS) display high-quality inpainting and strong adherence to the prompt. Images in the remaining quadrants reveal more nuanced characteristics. Specfically, in the top-left quadrant (low CLIP-S, high MPS), images may feature well-executed inpainting but low semantic alignment with the prompt. Interestingly, low CLIP-S in this quadrant appears to be influenced by large mask sizes relative to the requested content, as illustrated by examples such as *"Small flower"* and *"A black cat resting on the sidewalk"*. Finally, images in the bottom-right quadrant (high CLIP-S, low MPS) correspond to cases where the inpainted region aligns with the target prompt but exhibits visually unpleasant results.

These observations underscores the complexity of evaluating inpainting performance in our setting, which depends not only on the quality of the inpainted region

but also on the interaction between mask size and prompt content. After filtering, the pipeline can optionally regenerate discarded samples up to a predefined maximum number of attempts, thereby increasing the probability of obtaining valid instances and improving the overall reliability of the resulting dataset.

Regeneration can be performed by re-running the inpainting process with the same diffusion model, prompt, mask, and hyper-parameters, varying only the random seed at the inpainting stage, ensuring distributional consistency under a fixed data-generating configuration. To quantify the additional computational cost introduced by this process, we can model the number of attempts required to obtain a valid sample using a simple probabilistic estimation.

Let p denote the probability of success, i.e., the probability that a single generation is correct (with both CLIP-S and MPS exceeding their respective thresholds). Let X be the random variable representing the number of generations required to obtain the first successful sample. Assuming independent generations, X follows a geometric distribution with probability mass function:

$$P(X = k) = (1 - p)^{k-1} p, \;\; k = 1, 2, 3, ...$$

This expression indicates that the first $k-1$ attempts are failures, each occurring with probability $1-p$, whereas the k-th attempt is the first successful one, with probability p. The expected number of generation required to obtain the first success is therefore:

$$\mathbb{E}[X] = \frac{1}{p}$$

This means that, on average, $1/p$ generations are needed to obtain a correct sample.

Therefore,Table 4 reports the expected number of generations required for obtaining an image with both CLIP-S and MPS over thresholds, for each dataset–model configuration. Unsurprisingly, the expected number of generations $\frac{1}{p}$ depends both on the source dataset and the inpaiting model. Configurations producing higher-quality outputs, such as VISION with Fooocus or VISION with BrushNet (see Table 3), generally exhibit larger values of p, and therefore lower expected numbers of generations. For instance, for VISION with BrushNet, $1/p = 3.75$ indicates that about 3.75 generations are required on average to obtain one valid sample, implying a computational cost roughly three times higher than the baseline without regeneration.

Although the regeneration strategy increases the computational cost, the overall pipeline remains fully automatic and does not require additional human effort.

While this filtering strategy cannot guarantee that the remaining images achieve the best global appearance and semantic coherence with the original scene, it provides a practical tool for discarding images in which the inpainted region is both misaligned with the prompt and poor in quality, according to the MPS definition. These observations suggest that there is significant room for improvement in semantic quality assessment of inpainting models.

Table 4: Expected number of generations required to obtain a successful sample for each dataset–model configuration. Values report $1/p$, where p denotes the probability that a single generation satisfies both CLIP-S and MPS thresholds. Lower values indicate higher per-generation success probability and thus fewer attempts needed on average to obtain a valid sample.

	Fooocus	**SDv2**	**BrushNet**	**DreamShaper v8**
Flickr30k	7.96	6.02	4.22	5.20
VISION	3.86	4.78	3.75	5.23
FloreView	6.00	8.16	4.86	7.55

Analysis of Alignment with Human Perception

To obtain an initial indication of the relationship between human perception and the behavior of the automatic filtering metrics, we conducted a human evaluation. Specifically, we designed a small-scale pairwise perceptual quality test, in which participants were shown pairs of inpainted images side by side and asked to select the one with higher perceived visual quality. The test was intentionally designed to be simple and lightweight. In particular, the prompts used to generate the images were not shown to participants in order to reduce task complexity, avoid distracting users from the visual assessment, and keep the evaluation time short. For this evaluation, we focused exclusively on images with medium mask size and generated using Foocus (F) and BrushNet (B), as these models achieved the highest MPS scores (see Table 3). Each pair consisted of one "kept" image (i.e., an image whose MPS and CLIP-S scores exceeded the predefined thresholds) and one "discarded" image (i.e., an image whose MPS or CLIP-S scores fell below the thresholds). We considered two types of image pairs: 1) Same-content pairs, where both images originate from the same source image but differ in the applied prompt and/or inpainting model, 2) Different-content pairs, where images originate from different source images and also differ in the applied manipulation and/or inpainting model. We specify that in same-content pairs, images share same mask but different prompt, while in different-content pairs they differ in both mask and prompt. In total, we selected 52 kept–discarded pairs, uniformly distributed across the source datasets Flickr30k, VISION, and FloreView. The pairs were organized as: 9 different-content pairs (B vs. F), 9 different-content pairs (F vs. B), 3 same-content pairs (B vs. B), 3 same-content pairs (F vs. B), 3 same-content pairs (B vs. B), 9 different-content pairs (B vs. B), 9 different-content pairs (F vs. F), 9 same-content pairs (F vs. F). The test involved 121 users which accessed a dedicated web application with a unique token. Before starting the evaluation, participants were presented with an introductory explanation of the task, along with two example pairs that were not part of the actual test set. These examples were included to ensure a clear understanding of the evaluation procedure. Each pair was displayed for 10 seconds, resulting in a total session duration of approximately 10 minutes. To evaluate the alignment between the proposed automatic filtering stage and human perception, we

compute the agreement rate, as as a binary variable equal to 1 when the participant's choice matched the ground truth and 0 otherwise.

Overall, we observe a mean agreement of 0.48 between human preference and the automatic filtering decision. When restricting the analysis to same-content pairs, the agreement slightly increases to 0.51. In contrast, for different-content pairs, the agreement is 0.47. This difference can be explained by the nature of the two evaluation conditions. In the same-content setting, participants are naturally encouraged to focus on the manipulated region, which more closely reflects the behavior of the automatic metrics, as these are computed specifically on the inpainted area while ignoring the surrounding context. In the different-content condition, however, the evaluation becomes more subjective: participants tend to assess the image as a whole, and their judgment may also be influenced by properties of the original image, such as resolution or other visual artifacts. As a result, human preferences become less aligned with the metric-based evaluation.

Although these results may not appear entirely satisfactory, they were somewhat expected given the intended simplicity of our experimental setting, and highlight important differences between human and automatic evaluation criteria. In particular, automatic metrics evaluate only the inpainted region by design and explicitly account for prompt alignment, while human participants observe the entire image and do not have access to the prompt. Consequently, humans tend to prioritize the semantic coherence of the overall scene, while prompt adherence does not influence their judgment. This effect is illustrated in Figure 12, where the automatic filter selects the image with higher CLIP-S as high quality (HQ), whereas users often prefer the alternative because it does not introduce elements that alter the scene. Furthermore, artifacts near the boundaries of the inpainted region may affect the perceived quality of the whole image and influence human judgments, while remaining unaccounted for by metrics that evaluate only the inpainted area (see Figure 11). Examples of pairs with high agreement rate are represented in Figure 13.

In conclusion, although this human evaluation is not sufficient to provide statistically significant quantitative results, it offers useful insights into potential directions for improvement. These include both refinements to the evaluation design and the development of automatic metrics that better account for semantic coherence, thereby improving their alignment with human judgments.

4 Discussion and Implications

In this section, we summarize the key aspects of the proposed framework, emphasizing the elements that contribute most to its performance and usability.

The overall quality of the final outputs is largely influenced by the performance of each pipeline's component: the selection of the source dataset, the adopted segmentation workflow, the prompt generation mechanism, and the inpainting model. To better illustrate this characteristic, we considered the BtB dataset, including its Flickr30k, VISION, and FloreView subsets, as well as the OpenImages-100 dataset, obtained by using the same configuration for prompt generation and inpainting modules. Table 5 shows the percentage of remaining images for BtB and OpenImages-100 datasets after

applying the filtering criterion defined in 3.4, while Table 6 reports the corresponding average CLIP-S and MPS scores. As we can observe, OpenImages-100 is the dataset with the fewest filtered-out images and the highest average scores for both CLIP-S and MPS. This result strongly depends of the nature of the source dataset, as the prompt generation and the inpainting processes are the same across all datasets compared. Specifically, the authors of Open Images v7 dataset applied strict selection criteria to the images for segmentation annotation, retaining only the ones having well-defined label categories and excluding objects with complex shapes that are more challenging for neural networks to segment. They further discarded images with boxes smaller than 40×80 or 80×40 pixels, as well as those with blurry, dark boundaries or containing multiple objects. Consequently, the images equipped with segmentation masks belonging to Open Images v7, differs substantially from those originating BtB dataset, with a prevalence of simpler scenes with fewer subjects and refined masks, typically composed of a unique connected component, as shown in Figure 14. In particular, restricting to OpenImages-100, the 84% of the images include only one connected component.

By contrast, BtB datasets are obtained starting from RD-SAM segmentation workflow, which exhibits a high proportion of masks with multiple connected components, as shown in Table 1. This indicates a higher degree of fragmented masks, which are inherently more challenging to replace or represent with a single, well defined prompt compared to a single coherent object. Indeed, if we analyze the distribution of images with masks having a number of connected components greater than one in the BtB dataset, we find that only the 16% of those images fall within the top-right quadrant of Figure 10, corresponding to high-quality outputs according to our filtering criterion.

For this reason, it is preferable to choose a mask extraction workflow that yields a high rate of segmentation masks with a single connected component, such as SEEM and Florence-2, as this increases the likelihood of achieving higher-quality results in the final image generation process.

Regarding the prompt generation component, the choice of the VLM should be guided by its ability to produce responses that satisfy the defined constraints. Since prompt retrieval is an automated process, it is crucial to select a model that consistently adheres to the required template. As previously shown in Table 2, a suitable option might be InternVL3, which never produces responses in an incorrect format and generates excessively long outputs in only 2% of cases. Naturally, the selection also depends on the available hardware resources, as larger models typically require greater computational capacity.

Finally, among all components, inpainting represents the most challenging step to evaluate. The quality of the final output depends not only on the chosen inpainting model but also on the region targeted for manipulation and the corresponding prompt. Although choosing a strong inpainting tool, such as BrushNet or Fooocus (see Table 3), is essential, this is not sufficient on its own to guarantee high-quality results.

5 Conclusions

In this paper, we presented Beyond the Brush++, a fully automated and extensible pipeline designed to generate large-scale collections of realistic inpainted images.

Table 5: Percentage of remaining images for BtB subsets and OpenImages-100 dataset after applying the filtering criterion on CLIP-S and MPS.

	BtB-Flickr30k	**BtB-VISION**	**BtB-FloreView**	**OpenImages-100**
CLIP-S ≥ 0.61 & MPS ≥ 2.86	0.13	0.22	0.17	0.37

Table 6: Average CLIP-S and MPS scores for images inpainted following the BtB pipeline [11] from different source datasets

	BtB-Flickr30k	**BtB-VISION**	**BtB-FloreView**	**OpenImage-100**
Avg CLIP-S	0.57	0.58	0.58	0.64
Avg MPS	0.05	0.75	0.15	2.10

BtB++ improves the original BtB [11] by experimenting with different models across the various components of the pipeline (mask extraction, prompt generation, and inpainting) and by introducing an additional automatic filtering module to discard and, if necessary, regenerate low-quality images. Specifically, for mask extraction, we implemented a segmentation workflow that can be replaced by a segmentation model in combination with other foundation models, or with all-in-one approach that generates masks directly from the input image. For prompt generation, we employed four different Visual Language Models to produce diverse and semantically coherent textual descriptions for each masked area. In the inpainting stage, we leveraged four distinct diffusion models to create high-quality manipulations. Finally, the automatic filtering module evaluates each generated image using CLIPScore and Multi-dimensional Preference Scoring (MPS), discarding or regenerating low-quality outputs to ensure the overall quality of the resulting dataset.

We demonstrated BtB++'s flexibility and modularity through comparisons of different models for each component and showed that the pipeline can operate effectively even when individual modules are disabled, such as when masks are already available.

The image-quality filter provides an effective tool to discard outputs where the inpainted region poorly aligns with the prompt or exhibits low overall quality. However, it focuses only on the inpainted area and thus it does not guarantee that the remaining images are the most realistic, as the consistency between the inpainted object and the surrounding scene is not explicitly considered. The human quality test revealed a partial alignment between participants' judgments and the automatic filter, providing valuable insights into the differences between human perception and current metric-based evaluation, particularly regarding semantic coherence between the inpainted region and the surrounding scene.

Future work could enhance this filtering strategy by incorporating measures of semantic and visual coherence, ensuring that the generated regions integrate seamlessly with the rest of the scene. Additionally, extending the BtB++ pipeline to handle video inpainting would open new opportunities for creating large-scale, high-quality

datasets of temporally consistent manipulated content, which could be highly valuable for research in video forensics.

Declarations

- **Funding** None to declare
- **Conflict of interest/Competing interests** None to declare
- **Ethics approval and consent to participate** Not applicable
- **Consent for publication** Not applicable
- **Data availability** Not applicable
- **Materials availability** Not applicable
- **Code availability** Code will be released upon acceptance
- **Author contribution** Conceptualization, G.B., C.A., D.B., and D.S.; Methodology, G.B.; Software, G.B. and C.A.; Investigation, G.B.; Data Curation, G.B and C.A.; Writing-original draft, G.B. and C.A.; Writing—review and editing, G.B., C.A., D.B., D.S., and A.P.; Visualization, G.B. and C.A.; Supervision, D.B., D.S., and A.P.

References

[1] Hendrix, J., Morozoff, D.: Media forensics in the age of disinformation. In: Multimedia Forensics, pp. 7–40. Springer, Singapore (2022)

[2] Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 417–424 (2000)

[3] Mehrjardi, F.Z., Latif, A.M., Zarchi, M.S., Sheikhpour, R.: A survey on deep learning-based image forgery detection. Pattern Recognition **144**, 109778 (2023) https://doi.org/10.1016/j.patcog.2023.109778

[4] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4471–4480 (2019)

[5] Zheng, H., Lin, Z., Lu, J., Cohen, S., Shechtman, E., Barnes, C., Zhang, J., Xu, N., Amirghodsi, S., Luo, J.: Image inpainting with cascaded modulation gan and object-aware training. In: European Conference on Computer Vision, pp. 277–296 (2022). Springer

[6] Liu, H., Wang, Y., Qian, B., Wang, M., Rui, Y.: Structure matters: Tackling the semantic discrepancy in diffusion models for image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8038–8047 (2024)

[7] Ko, K., Kim, C.-S.: Continuously masked transformer for image inpainting. In:

Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13169–13178 (2023)

[8] Mahfoudi, G., Tajini, B., Retraint, F., Morain-Nicolier, F., Dugelay, J.L., Marc, P.: Defacto: Image and face manipulation dataset. In: 2019 27Th European Signal Processing Conference (EUSIPCO) (2019). IEEE

[9] Novozamsky, A., Mahdian, B., Saic, S.: Imd2020: A large-scale annotated dataset tailored for detecting manipulated images. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, pp. 71–80 (2020)

[10] Wang, S., Saharia, C., Montgomery, C., Pont-Tuset, J., Noy, S., Pellegrini, S., Onoe, Y., Laszlo, S., Fleet, D.J., Soricut, R., *et al.*: Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18359–18369 (2023)

[11] Bertazzini, G., Albisani, C., Baracchi, D., Shullani, D., Piva, A.: Beyond the brush: Fully-automated crafting of realistic inpainted images. In: 2024 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6 (2024). IEEE

[12] Giakoumoglou, P., Karageorgiou, D., Papadopoulos, S., Petrantonakis, P.C.: Sagi: Semantically aligned and uncertainty guided ai image inpainting. arXiv preprint arXiv:2502.06593 (2025)

[13] Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 7514–7528 (2021)

[14] Zhang, S., Wang, B., Wu, J., Li, Y., Gao, T., Zhang, D., Wang, Z.: Learning multi-dimensional human preference for text-to-image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8018–8027 (2024)

[15] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., *et al.*: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)

[16] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., *et al.*: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. International journal of computer vision **128**(7), 1956–1981 (2020)

[17] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G.,

Askell, A., Mishkin, P., Clark, J., *et al.*: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021). PmLR

[18] Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics **2**, 67–78 (2014)

[19] Shullani, D., Fontani, M., Iuliani, M., Shaya, O.A., Piva, A.: Vision: a video and image dataset for source identification. EURASIP Journal on Information Security **2017**, 1–16 (2017)

[20] Baracchi, D., Shullani, D., Iuliani, M., Piva, A.: Floreview: an image and video dataset for forensic analysis. IEEE Access (2023)

[21] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023)

[22] Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., et al.: Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714 (2024)

[23] Zhang, Y., Huang, X., Ma, J., Li, Z., Luo, Z., Xie, Y., Qin, Y., Luo, T., Li, Y., Liu, S., et al.: Recognize anything: A strong image tagging model. arXiv preprint arXiv:2306.03514 (2023)

[24] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)

[25] Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a unified representation for a variety of vision tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4818–4829 (2024)

[26] Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., Wang, L., Gao, J., Lee, Y.J.: Segment everything everywhere all at once. Advances in neural information processing systems **36**, 19769–19782 (2023)

[27] Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., *et al.*: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 24185–24198 (2024)

[28] Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian,

H., Liu, Z., et al.: Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. arXiv preprint arXiv:2412.05271 (2024)

[29] Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., Li, B., Luo, P., Lu, T., Qiao, Y., Dai, J.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. arXiv preprint arXiv:2312.14238 (2023)

[30] Chen, Z., Wang, W., Tian, H., Ye, S., Gao, Z., Cui, E., Tong, W., Hu, K., Luo, J., Ma, Z., et al.: How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. arXiv preprint arXiv:2404.16821 (2024)

[31] Team, Q.: Qwen2.5-VL (2025). https://qwenlm.github.io/blog/qwen2.5-vl/

[32] Marafioti, A., Zohar, O., Farré, M., Noyan, M., Bakouch, E., Cuenca, P., Zakka, C., Allal, L.B., Lozhkov, A., Tazi, N., Srivastav, V., Lochner, J., Larcher, H., Morlon, M., Tunstall, L., Werra, L., Wolf, T.: Smolvlm: Redefining small and efficient multimodal models. arXiv preprint arXiv:2504.05299 (2025)

[33] Zhang, L.: Fooocus (2023). https://github.com/lllyasviel/Fooocus

[34] Fooocus-API (2023). https://github.com/mrhan1993/Fooocus-API

[35] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10684–10695 (2022)

[36] Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2149–2159 (2022)

[37] Ju, X., Liu, X., Wang, X., Bian, Y., Shan, Y., Xu, Q.: Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In: European Conference on Computer Vision, pp. 150–168 (2024). Springer

Fig. 8: Examples from the BtB-Flickr30k subset with corresponding CLIPScores. The top row shows successful manipulations with high CLIPScores (above 0.6), where the requested prompt is clearly integrated into the original scene. The bottom row shows unsuccessful cases with low CLIPScores (below 0.6), where the prompt was not incorporated. Manipulated regions are highlighted in red.

Fig. 9: Examples from the OpenImages-100 dataset with corresponding MPS scores. The top row shows high-quality manipulations (with MPS above 2.86), while the bottom row shows low-quality results (with MPS below 2.86). Manipulated regions are highlighted in red.

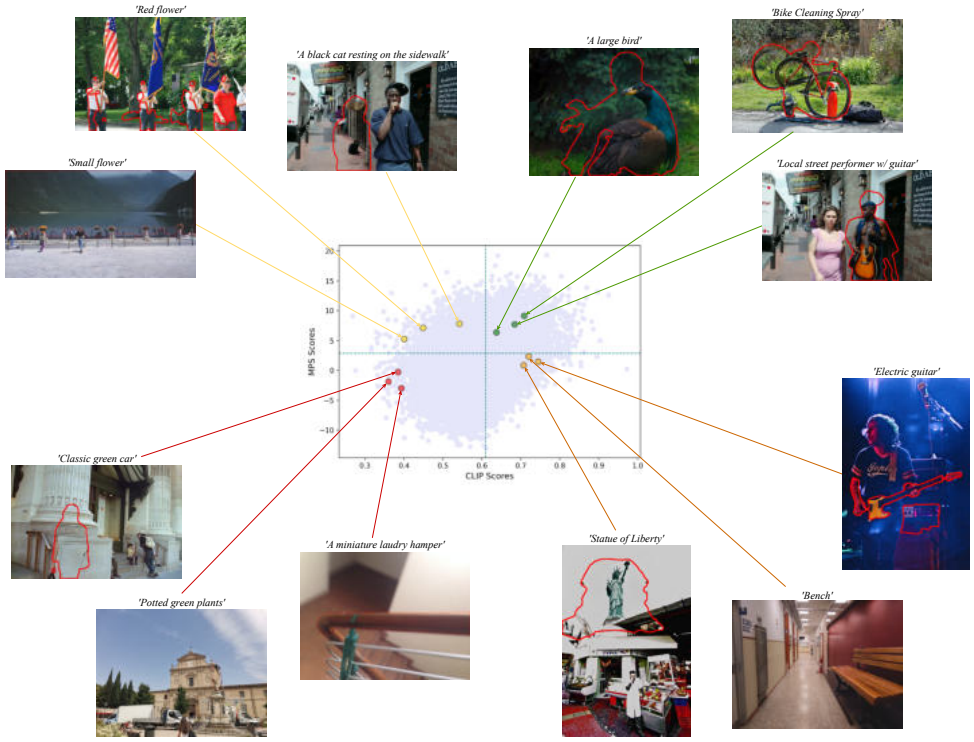

Fig. 10: MPS vs CLIPScores on BtB dataset. Teal dashed lines refers to 70th percentile of MPS (horizontal line) and CLIP (vertical line) scores.

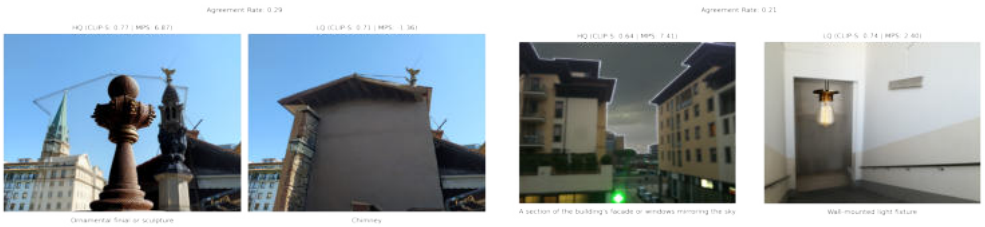

Fig. 11: Examples of pairs with low agreement, where high quality (HQ) images contain artifacts at the inpainted region boundaries.

Fig. 12: Examples of pairs with low agreement, where the high quality (HQ) image is primarily selected based on high CLIP-S.

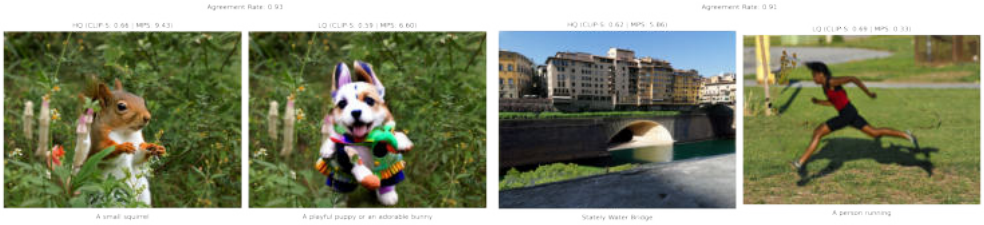

Fig. 13: Examples of pairs with high agreement rates.

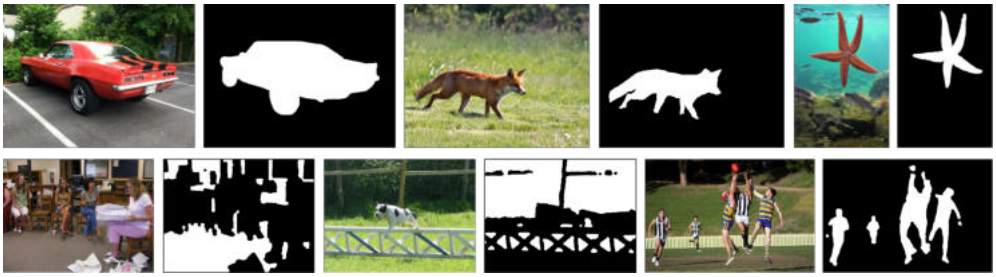

Fig. 14: Real images and corresponding masks from Open Images v7 (top row) and Flickr30k (bottom row) datasets.