MULAN: A Multi Layer Annotated Dataset for Controllable Text-to-Image Generation

Petru-Daniel Tudosiu*[1] Yongxin Yang[1] Shifeng Zhang[1] Fei Chen[1]
Steven McDonagh[2 †] Gerasimos Lampouras[1] Ignacio Iacobacci[1] Sarah Parisot*[1]

[1]Huawei Noah's Ark Lab, [2]University of Edinburgh

Abstract

Text-to-image generation has achieved astonishing results, yet precise spatial controllability and prompt fidelity remain highly challenging. This limitation is typically addressed through cumbersome prompt engineering, scene layout conditioning, or image editing techniques which often require hand drawn masks. Nonetheless, pre-existing works struggle to take advantage of the natural instance-level compositionality of scenes due to the typically flat nature of rasterized RGB output images. Towards adressing this challenge, we introduce MuLAn: a novel dataset comprising over 44K MUlti-Layer ANnotations of RGB images as multi-layer, instance-wise RGBA decompositions, and over 100K instance images. To build MuLAn, we developed a training free pipeline which decomposes a monocular RGB image into a stack of RGBA layers comprising of background and isolated instances. We achieve this through the use of pre-trained general-purpose models, and by developing three modules: image decomposition *for instance discovery and extraction, instance* completion *to reconstruct occluded areas, and image* re-assembly. *We use our pipeline to create MuLAn-COCO and MuLAn-LAION datasets, which contain a variety of image decompositions in terms of style, composition and complexity. With MuLAn, we provide the first photorealistic resource providing instance decomposition and occlusion information for high quality images, opening up new avenues for text-to-image generative AI research. With this, we aim to encourage the development of novel generation and editing technology, in particular layer-wise solutions. MuLAn data resources are available at* https://MuLAn-dataset.github.io/

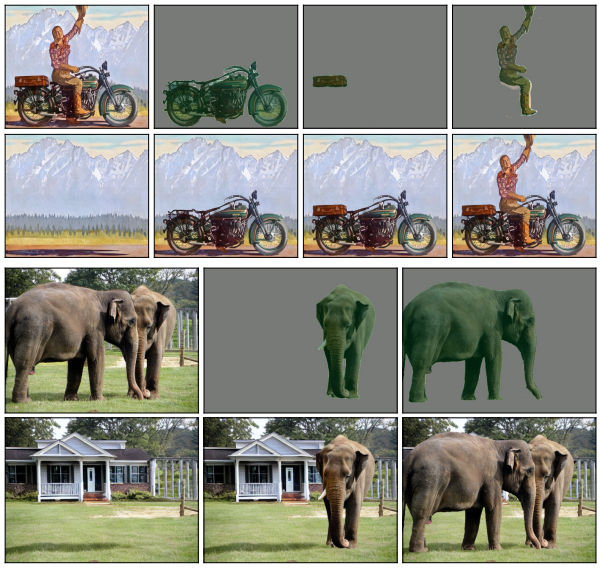


Figure 1. Example annotations from our MuLAn dataset. We decompose an image into a multi-layer RGBA stack, where each layer comprises an instance image with transparent alpha layer (green overlays) and background image. For each scene, the second row shows iterative addition of RGBA instance layers.

1. Introduction

Large scale generative diffusion models [27, 31] now enable generation of high quality images from text prompt descriptions. These models are typically trained on large datasets of captioned RGB images encompassing multiple styles and contents. While such techniques have pushed the field of text-guided image generation forward tremendously, precise controllability of image appearance and composition (*e.g.* local image attributes, countability) still remains a challenge [14]. Prompt instructions can often lack precision or be misunderstood (*e.g.* counting errors, incorrect spatial positions, bleeding of concepts, failure to add or remove instances), and therefore require intricate prompt engineering to obtain the desired result. Fine-tuning a generated image by even slightly changing the prompt can result in a markedly different sample, further increasing the amount of effort required to obtain the desired image.

* equal contribution, † work done in part at Huawei Noah's Ark Lab

Efforts towards addressing these limitations have considered additional conditioning in the form of *e.g.* poses, segmentation maps, edge maps [26, 40] and model-based image editing strategies [5, 8, 12]. The former improves spatial controllability, yet still requires tedious prompt engineering to adjust image content; while the latter often fails to understand spatial instructions and therefore struggles to accurately modify desired image regions without affecting other areas or introducing unwanted morphological changes.

We conjecture that a key obstacle is the typically flat nature of rasterised RGB images, which fails to leverage of the compositional nature of scene content. Alternatively, isolating instances and background on individual RGBA layers has the potential to grant precise control over image composition, as processing of instances on separate layers guarantees content preservation. This can trivialise image manipulation tasks like resizing, moving, or adding/removing elements, which remain a challenge for current editing methods.

Collage Diffusion [32] and Text2Layer [41] have shown preliminary evidence of the benefits of multi-layer composable image generation. Collage diffusion controls image layout by composing arbitrary input layers *e.g.* by sampling composable foreground and background layers, while Text2Layer explores decomposition of images into two separate layers (grouping foreground instances and background). Despite an increasing interest in training-free layered and composite generation [19, 23], a major barrier to research development in this promising direction is the lack of publicly available photorealistic, multi-layer data to train and evaluate generative and editing methodology.

In this work, we aim to fill this gap by introducing MuLAn, a novel dataset comprising of multi-Layer RGBA decomposition annotations of natural images (see Fig. 2 for an RGBA decomposition illustration). To achieve this, we design an image processing pipeline that takes as input a single RGB image and outputs a multi-layer RGBA decomposition of its background and individual object instances. We propose to leverage large-scale pre-trained foundational models to build a robust, general purpose pipeline without incurring any additional model training costs. We separate our decomposition process into three submodules, focusing on 1) instance discovery, ordering and extraction, 2) instance completion of occluded appearance, and 3) image re-assembly as an RGBA stack. Each submodule is carefully designed to ensure general applicability, high instance and background reconstruction quality, and maximal consistency between input image and composed RGBA stack. We process images from the COCO [20] and LAION Aesthetics 6.5 [33] datasets using our novel pipeline, yielding multi-layer instance annotations for over 44K images and over 100K instances. Illustrations of generated decompositions are shown in Fig. 1: each decomposed image comprises a background layer, and extracted instances are separate RGBA images with transparency alpha layers. Instances can be removed from the RGBA stack, yielding several intermediate representations; where resulting occluded areas are completed via inpainting.

Dataset	# Images	Resolutions	# Classes	# Instances	Occluded Instances	Average Occlusion Rate	Synthetic	Ordering
SAIL-VOS [13]	111,654	800x1280	162	1,896,296	1,653,980	56.3 %	√	√
OVD [35]	34,100	500x375	196 (vehicles)	-	-	-	√	-
WALT [30]	15 Mil	4K/1080p	2 (vehicles)	36 Mil	-	-	partially	√
AHP [43]	56,599	Non-fixed	1 (humans)	56,599	-	-	partially	-
DYCE [7]	5,500	1000x1000	79 (indoor scenes)	85,975	70,766	27.7%	√	-
OLMD [6]	13,000	384x512	40 (indoor scenes)	-	-	-	√	√
CSD [42]	11,434	512x512	40 (indoor scenes)	129,336	74,596	26.3%	√	√
MuLAn-COCO	16,034	Non-fixed	662	40,335	15,223	7.2 %	partially	√
MuLAn-LAION	28,826	Non-fixed	705	60,934	14,009	8.2 %	partially	√
MuLAn	44,860	Non-fixed	759	101,269	29,232	7.7 %	partially	√

Table 1. Comparison between MuLAn and publicly available related datasets providing amodal masks and appearance information.

Our goal in releasing MuLAn, is to foster development and training of technologies to generate images as RGBA stacks, by offering comprehensive scene decomposition information and scene instance consistency. We aim to facilitate research seeking to (i) advance controllability of generated image structures and (ii) improve local image modification quality, via precise layer-wise instance editing. This paper illustrates the potential utility of our dataset and the benefits of layer-wise representations through two applications: 1) RGBA image generation and 2) instance addition image editing. In summary, our main contributions are:

- The release of MuLAn, a novel dataset of multi-layer annotations, comprising RGBA decompositions of over 44K images, derived from COCO and LAION Aesthetics 6.5. To the best of our knowledge, MuLAn is the first dataset of its kind, providing instance decomposition and occlusion information for a large variety of scenes, styles (including photorealistic real images), resolutions and object types.
- A novel, modular pipeline that decomposes single RGB images into instance-wise RGBA stacks at no additional training cost. Our idea leverages large pre-trained models in an innovative manner, and comprises ordering and iterative inpainting strategies to achieve our image decomposition objective. This further enables unique insight into the behaviour of popular large-models in the wild.
- We showcase MuLAn's potential through two applications that leverage our rich annotations in distinct ways.

2. Related Work

Amodal completion aims to automatically estimate the real structure and appearance of partially occluded objects. This challenging task has been heavily researched [1], typically building models that are trained on synthetic or richly annotated datasets. Such datasets typically comprise instance segmentation masks that include occluded regions. In addition, the closest datasets to MuLAn comprise appearance information of occluded areas and instance ordering information. We provide a detailed comparison of these datasets with ours in Tab. 1. Time and cost requirements of produc-

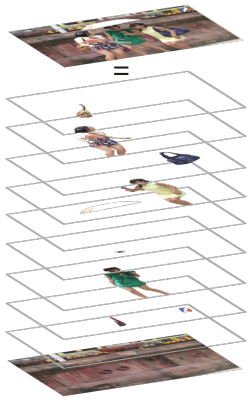


Figure 2. Illustration of our RGBA decomposition objective.

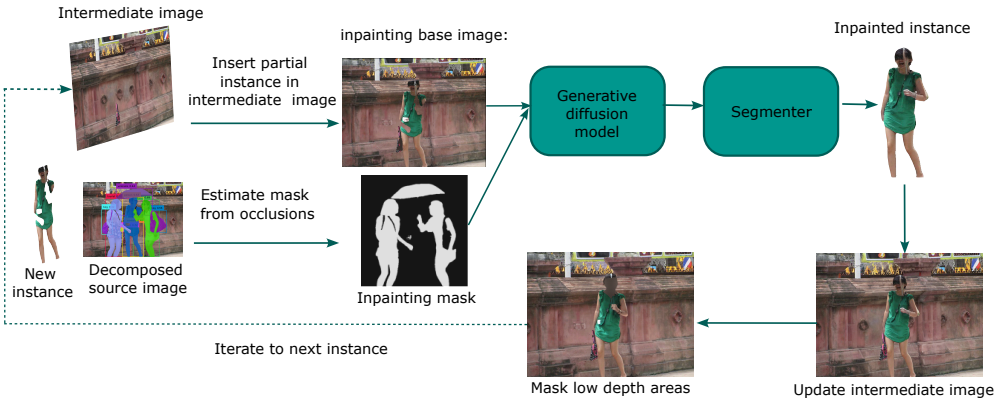


Figure 3. Illustration of the inpainting procedure for a given instance.

ing ground truth amodal annotations have limited previous research to synthetic, small or highly specialised datasets such as indoor scenes [6, 7, 42], humans [43], vehicles [35], and objects and humans [13, 30]. In contrast, MuLAn comprises images with a large variety of scenes, styles (including photorealistic real images), resolutions and object types; and was built on top of popular datasets to support generative AI research. We highlight that our use of real images impacts occlusion rate compared to existing datasets, where synthetic scenes were purposely designed to have a high rate.

RGBA image decomposition requires identifying and isolating image instances on individual transparent layers, and estimating the shape and appearance of occluded areas. This challenging task is typically carried out using additional inputs (beyond a single RGB image), such as inmodal segmentations [39], stereo images [9] and temporal video frames. The latter substantially facilitates the decomposition tasks, as video frames can provide missing occluded information [22, 34]. Recently, layer based generative modelling benefits from initial explorations. Text2Layer [41] creates a two-layer RGBA decomposition of natural images. Images are decomposed into a background and salient foreground layer, where the background is inpainted using prompt-free state-of-the-art diffusion models. The main limitation of this, compared to our approach, is the two-layer decomposition: all instances are extracted in the same foreground layer which critically lacks our required flexibility of instance wise decomposition. Our objective, to decompose each instance individually, comes with additional challenges such as instance ordering, instance inpainting and amodal completion. Adjacent to our decomposition objective, PCNet [39] learns to predict instance ordering, amodal masks and object completion. The approach's applicability is however restricted by the aforementioned limitations of amodal completion datasets. To the best of our knowledge, our decomposition pipeline is the only general purpose technology capable of decomposing monocular RGB images.

Complementary to our work, Collage Diffusion [32], an image collage strategy for diffusion generative models, is developed with a similar instance-level modularity objective. While we aim to extract instances from an image, their method seeks to assemble individual instances in a homogeneous composite image. One limitation of this prior work involves the challenge of balancing appearance preservation of collaged instances with homogeneity of the composite image, which can be considered non-trivial and increases in difficulty with instance count.

3. Image Decomposition Pipeline

Our objective is to decompose a single RGB image I into an instance-wise stack of N RGB-A image layers $\mathcal{S} = \{l_i \,|\, i \in 1, \ldots, N\}$, where the A-layer (Alpha) describes the transparency of each RGB instance. As illustrated in Fig. 2, each layer l_i comprises either a background image, or a single instance with full transparency in non-instance regions. Flattening S should yield our original image I. Due to the lack of large, general purpose datasets to provide this level of granular information, we approached this objective in a training free manner, leveraging a combination of specialised large scale pre-trained models.

We build a pipeline that comprises a sequence of three main modules. First, the *Decomposition* module encompasses instance discovery and extraction. It focuses on scene understanding, comprising a sequence of object detection, segmentation and depth estimation models. This module decomposes I into a background image and a series of extracted instances with associated segmentation masks. At this stage, background and extracted instances are missing information due to occlusions. This is addressed with our second main module: *Instance Completion*. This second component aims to reconstruct each instance individually, as it would look like if there were no occlusions. This step

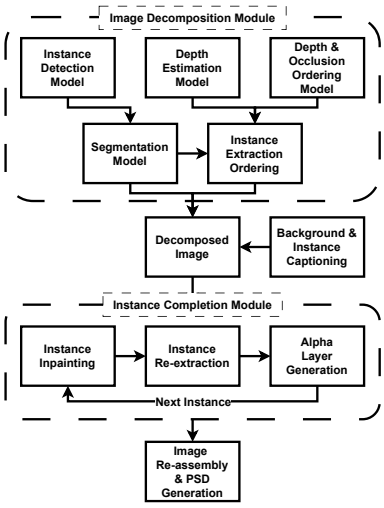


Figure 4. Overview of our RGBA decomposition pipeline

leverages depth and relative occlusion information to establish an instance ordering, and state-of-the-art text-to-image generative models to inpaint occluded areas. Finally, the image *Reassembly* module generates occlusion aware Alpha layers, and builds our RGBA stack such that flattening the stack effectively reconstructs I.

An overview of our pipeline is found in Fig. 4 and a further detailed schematic, showing all components' instantiations, is also available in the supplementary materials.

3.1. Image Decomposition Module

Our decomposition module aims to extract and isolate all instances in the image. We first identify and segment instances using object detection and segmentation models. In parallel, we rely on depth estimation and occlusion ordering models to build relative occlusion graphs, and establish an instance ordering for extraction, inpainting and reassembly.

Object Detection. Accurately detecting all relevant instances in an image is the first step of our pipeline. In order to achieve good quality decomposition, it is essential that we are able to detect and separate all instances present in the scene. To this end, we leverage vision-language object detection techniques, that input a list of categories to detect, alongside the input image. Such models are attractive as they easily allow open-set detection, meaning we are not limited to the pre-existing class sets of specific data. We use detCLIPv2 [37, 38], a state-of-the-art model characterised by its ability to leverage category definitions (and not just class names) to improve detection accuracy. We carefully construct our text input (the category list), to ensure all desired categories are detected and extracted from the image. We use the concept list from the THINGS [11] database, and manually update and simplify it to obtain more generic category names (*e.g.* merging types of boats, drinks, nuts, etc.), and remove homonyms and concepts that we do not want to extract (*e.g.* unmovable objects, clothing, bolts and hinges). We highlight that this list constitutes an input to the pipeline, easily allowing customisation of which instances to detect. This class list is used, alongside definitions from the WordNet [24] database, to identify all relevant instances in an image. This step of the pipeline outputs a series of bounding boxes with corresponding category names.

Segmentation. Our next step is to precisely segment detected instances. In order to handle a large number of categories, domains and image qualities we seek to leverage a robust general purpose segmentation model. One such model is SAM [16], which has been trained with the required diversity and scale, achieving good robustness and transferability across a large set of domains. The ability to use bounding boxes as grounding for segmentation predictions makes this family of models an excellent candidate to be combined with our detCLIPv2 detector.

Depth Estimation. Understanding the relative positioning of instances in an image is crucial towards achieving our RGBA decomposition goal. Depth estimation provides essential information, indicating distance to the camera at capture time. We use the MiDaS model [29], chosen for its robustness: it was trained on 12 different datasets allowing it to be reliable across different type of scenes and image qualities. Once computed, we quantise the depth map into multiple bins of width 250 relative depth units to facilitate cross instance comparisons.

Instance Extraction. We define instance extraction as the application of a binary mask onto the full image in order to isolate a detected instance from the rest of the image. We employ a set of strategies to increase the robustness of this crucial step. First, we estimate a proto-ordering by clustering instances based on their bounding box overlap, and use bounding box size and mean depth value (within the segmentation mask) to order them. Second, we use our proto-ordering to enforce disjoint instance segmentation masks by excluding extracted areas from the masks of following instances. Lastly, instances whose largest connected component is smaller than 20 pixels or 0.1% of the entire image are not extracted.

Depth and Occlusion Graphs. We further compute depth and occlusion instance graphs for a more comprehensive scene understanding. Specifically we are using the InstaDepthNet^{o,d} model [17], which is capable of jointly predicting occlusion and depth relationships between instances. The model predicts instance pairwise relationships, using the original image and instance segmentations as input. The directed occlusion graph outlines relative occlusion information between instances. Image instances are represented by graph nodes, and a directed edge from instance A to instance B (A $\rightarrow$ B) indicates that A occludes B. We note that valid bidirectional edges exist where two instances mutually

occlude one another (A $\leftrightarrow$ B). Similarly, the directed depth graph also represents instances as nodes, and A $\rightarrow$ B indicates that A is closer to the camera than B, as defined by instance mean depth values. A bidirectional edge (A $\leftrightarrow$ B) indicates that both instances have the same mean depth.

Instance ordering. In order to maximise instance completion quality, inpainting of occluded areas is performed using contextual information from the original image. As a result, establishing a precise instance inpainting schedule is crucial towards progressively enriching the image context without occluding relevant areas. We generate our instance ordering in three steps, relying on depth ordering and occlusion information obtained in our decomposition step. First, instances are ordered based on their depth information, from furthest away to closest (according to instance mean depth value). This can easily be achieved using the instance depth graph, by computing node out-degree: this computes the number of directed edges departing a node, *i.e*. the number of instances that are behind our node. Second, we rely our occlusion graph to refine our ordering: if instance A occludes instance B, instance B will systematically be ordered before instance A. Finally, mutually occluded instances are reordered according to their maximum depth value. Instance ordering algorithm details are provided in Supplementary materials.

3.2. Instance Completion Module

Prior to instance completion, we have successfully detected, isolated and ordered all instances from the background image. An important challenge remains: reconstructing occluded areas for each image layer l_i individually (including the background), such that removing or hiding any layer reveals occluded areas. Since we are decomposing natural images, this information is not available to us. We rely on state-of-the-art generative models to imagine these occluded areas from available context, using inpainting.

Diffusion model based image inpainting has set a new standard in comparison with traditional inpainting techniques [4, 15] as they take advantage not only of image contents but also of a learned image prior and textual conditioning. Even so, our setting comes with unique difficulties: 1) in contrast with the common strategy of carefully engineering hand-crafted prompts, we can only rely on automatically generated captions, 2) instance images comprise the instance on a background of uniform colour, an image mode not commonly seen by these models, and 3) rather than obtaining beautiful or creative images, we seek the simple, accurate and high quality completion for our available content. We next provide detail on our inpainting process and how these difficulties are addressed.

Inpainting procedure. An overview of our inpainting process is illustrated in Fig. 3. Given a pre-defined instance ordering, we iteratively inpaint an instance's occluded areas, starting from the background image to the closest instance. For a given instance, our inpainting process proceeds as follows: we first estimate an inpainting mask using occlusion ordering information and segmentation masks from occluding instances. Second, we build a contextual inpainting image by re-integrating our incomplete instance in an intermediate background image. This background image contains inpainted instances processed in previous iterations. Third, the instance is inpainted using a state-of-the-art inpainting generative model and automatic captions as prompts. Fourth, we re-extract the completed instance using our segmentation model and the occluded segmentation mask as guidance, effectively obtaining the complete instance image which will be part of our multi-layer representation. Finally, we update our background inpainting image for the next iteration by integrating our newly inpainted instance. Importantly, we aim to strike a balance between maximising scene context and preventing introduction of irrelevant image content. This is particularly important for mutually occluded instances: for example, considering a person holding a phone in their hands, with the person's hand as context, fingers will be reconstructed when inpainting the phone's occluded areas. To prevent this, we "hide" potentially misleading context by replacing information from pixels that have higher depth than the next instance's max depth with a constant value.

Inpainting mask. Estimating an accurate inpainting mask, *i.e*. describing which image regions will be overwritten, is crucial towards achieving accurate instance completion. Failing to include key occluded areas risks yielding incomplete results, while a mask that is too large risks altering the original image appearance. Ideally, one would estimate an accurate complete instance shape via amodal completion techniques [1]. Existing methods tend, however, to be dataset or object class specific with limited generalisation ability. We alternatively propose to leverage instrinsic biases of large generative models by providing a large inpainting mask, encompassing the area where the occluded object *could* be present. This is achieved by building an inpainting mask comprising segmentation masks of all occluding instances.

Inpainting prompts remain simple, as we seek a fully automated decomposition strategy. For instance inpainting, we leverage automatically generated instance captions (see Sec. 3.4). To inpaint the background image, we use a simple generic prompt (*"an empty scene"*) that ensures the generated inpainting background is as simple as possible. Importantly, we include class names of all other instances in all negative prompts, to avoid re-introducing extracted instances. This increases robustness to imperfect segmentations.

3.3. Image re-assembly module

The last and simplest module re-assembles all individual RGB images into an ordered RGBA stack that, once flattened, yields an image as close as possible to the the original input image. Instance RGB images are ordered following

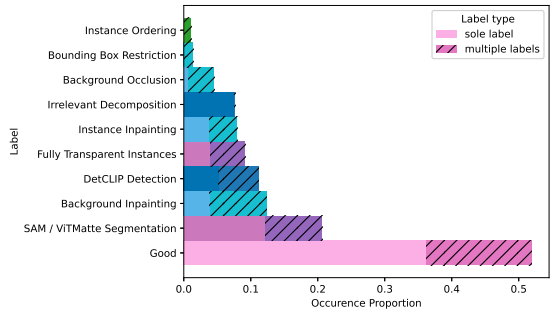


Figure 5. Failure distribution on manually annotated data subset.

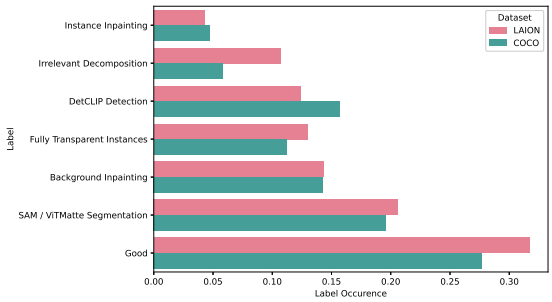


Figure 6. Failure distribution on automatically annotated data.

our inpainting ordering, such that instances inpainted last are at the top, with the background at the bottom of the stack. Following this order, we iteratively generate Alpha layers for each stack element by refining instance segmentation masks. We post process SAM segmentations obtained after inpainting with the image matting model VitMatte [36] to improve alpha blending quality, handle transparent objects, and address SAM undersegmentation tendencies. While undersegmenting is preferred for the first two modules, in order to avoid introducing proximal content and erroneous priors when inpainting, we require accurate delineations for this last stage. VitMatte refines SAM outputs, providing smoother non-binary segmentations, and allows us to blend the inpainted instances in a more natural way. In settings where mutual occlusion exist (*i.e*. a lower level instance is creating an occlusion), we further adjust alpha layers by setting occluded areas as transparent. This last module finally outputs our RGBA stack image decomposition.

3.4. Captioning strategy

We generate captions for all layers (background, instance), intermediate flattened RGBA stacks and the full image. We use LLaVa [21] to generate detailed captions for standard images. Due to the unique nature of instance images (an instance on a uniform white background), verbose captioning models like LLaVa tend to hallucinate image features. To address this, we leverage a BLIP-2 model [18] to caption instances and performed a grid search to select a parameter set limiting verbosity and hallucination. Furthermore, we use constricted beam search to generate multiple captions and select the best one with CLIP [28]. Components captioned with LLaVa are also captioned with BLIP, for completeness.

4. MuLAn Dataset

4.1. Base Datasets

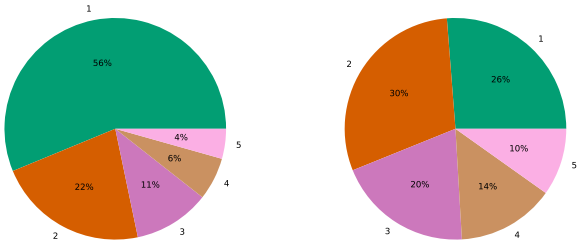
We run our full method on two datasets that provide sufficient scene compositionality to fully exploit our pipeline: the COCO [20] dataset and the Aesthetic V2 6.5 subset of the LAION [33] dataset. The Aesthetic subset filters the complete LAION dataset, selecting only images with an Aesthetic score of at least 6.5 and encompasses 625K images. To limit scene complexity and facilitate inspection, we only consider images that comprise one to five instances, which we determine using our object detector's output. We process all COCO images (58K images), and a random subset of 100K LAION images to limit computational cost.

4.2. Data Curation

We aim to build a dataset comprising high quality decompositions, and exclude potential failure modes. To this end, we manually inspect and label our processed data, identifying six main causes for decomposition failure:

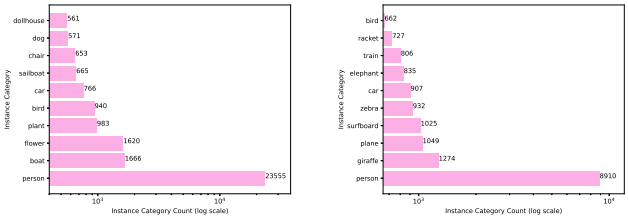
- **Object detection**: missing a key instance in an image, or multiple detections of the same object.
- **Segmentation**: incorrect instance segmentation on the original image, or after inpainting.
- **Background inpainting**: erroneous inpainting of the background image. This can be caused by imperfect segmentations, and our pipeline not accounting for causal visual instance effects on the scene (*e.g*. shadows).
- **Instance inpainting**: incorrect or incomplete inpainting of an instance. This often happens due to mask shape or pose biases (*e.g*. person holding a guitar).
- **Truncated instances**: image matting overly eroding the alpha mask of very small instances.
- **Irrelevant decomposition**: scenes that are not suited for instance-wise decompositions (*e.g*. scenes where part of the landscape was incorrectly detected).

Additionally, for analysis purposes, we annotate examples where the instance ordering is incorrect, where background elements occlude instances, and where instance completion is restricted by our bounding box constrained re-segmentation. We provide visual examples of failure modes in Supp. Materials. Using Voxel FiftyOne [25], we annotate 5000 randomly selected images from our processed pool of LAION Aesthetic 6.5 images, adding the 'good' label for successful decompositions. To mitigate biases, annotations were carried out independently by 3 annotators. We highlight that multiple labels can be assigned to a single image, and



(a) MuLAn-LAION (b) MuLAn-COCO

Figure 7. Scene distribution of MuLAn-LAION and MuLAn-COCO datasets. Distribution of number of categories per image.



(a) MuLAn-LAION (b) MuLAn-COCO

Figure 8. Top 10 most common categories in MuLAn subsets.

notably associate the 'good' label with other labels, when defects are minor and do not affect the overall validity of the decomposition. The distribution of failure modes over this manually annotated set is shown in Fig. 5, highlighting an overall success rate of 36% (52% with minor defects). We can see that segmentation issues are the biggest failure mode, followed by inpainting and object detection. Failures of our novel ordering, together with bounding box restrictions and background occlusion were the rarest issues.

Following [41], we leverage our manual annotations to train two classifiers to automatically annotate the rest of our processed data: an image level classifier flagging background and irrelevant decomposition issues, and an instance level multi-label classifier identifying remaining failure modes. Details on our classifier architectures and training process are discussed in Supp. Materials. Fig. 6 show the resulting label distribution for both LAION and COCO datasets. We adopt a conservative approach and select images with *only* a confident 'good' label as successful decompositions, and report only this portion of 'good' labels in Fig. 6. This yields 16K decompositions from COCO, and 28.9k from LAION, for a total of 44.8K annotations in our MuLAn dataset. Our automated failure modes distribution for LAION is very similar to our manually annotated portion, with segmentation and inpainting consistently the prominent issues. COCO follows a similar distribution, with larger object detection errors. This is expected as COCO is well known to be a challenging object detection benchmark (with COCO [20] and LVIS [10] annotations), with complex scenes. In contrast, LAION comprises simpler scenes with fewer instances.

4.3. Dataset Analysis

With our curated high quality annotations, we further analyse scene distribution and diversity of our 44.8K annotated images. Fig. 7 shows scene distribution in MuLAn in terms of number of instances per image. We can see that the LAION dataset has a majority of single instance images, which can be linked to the fact that highly aesthetic images tend to be simple scenes (*e.g.* portraits - this is also highlighted in Supplementary Fig. **??**). Nonetheless, MuLAn-LAION does contain sufficiently complex scenes, with 21% ($\simeq$ 6K) of images having more than three instances per images. MuLAn-COCO achieves good scene diversity, with 10% of the dataset comprising five instances, almost half of the dataset (44% $\simeq$ 7K) comprising more than three instances and only 28% ($\simeq$ 4.5K) of single instance images.

Next, we investigate scene diversity in terms of instance types. Out of the 942 detection categories, we obtain 662 and 705 categories in MuLAn-COCO and MuLAn-LAION, respectively, with a total of 759 categories in MuLAn. Fig. 8 shows the top ten most common categories in each dataset. While the *person* class is the majority class for both, it is overwhelmingly dominant in LAION. Besides persons, MuLAn-LAION mainly comprises inanimate and decor objects, while COCO comprises more active scenes, notably with animals and sports. Of the top ten categories, only three are common to both datasets (person, car and bird). These results highlight the complementarity of both dataset subsets, with MuLAn-LAION focusing on simpler, high quality and visually pleasing scenes, while MuLAn-COCO showcases more diverse scene types. The complete, sorted, list of categories per sub-dataset is available in Supp. Materials.

Finally, Fig. 12 presents additional visual examples of RGBA decompositions from MuLAn, showcasing a variety of scene compositions, styles and category types. Additional examples are available in Supp. Materials.

4.4. Dataset Applications

To illustrate the potential utility of our MuLAn dataset, we provide two experiments showcasing distinct example scenarios, under which our dataset can be leveraged.

RGBA Image generation. Our first application leverages MuLAn instances to adapt a diffusion model to generation of images with a transparency channel, by finetuning both the VAE and Unet of the Stable Diffusion (SD) v1.5 [31] model. In Fig 9, we provide visual comparisons of generated images, that are obtained using SD v1.5 with "on a black background" appended to the prompt and finetuned on our dataset in comparison with a model finetuned on 15,791 instances from multiple matting datasets. We can see that our dataset allows to generate better quality RGBA instances,

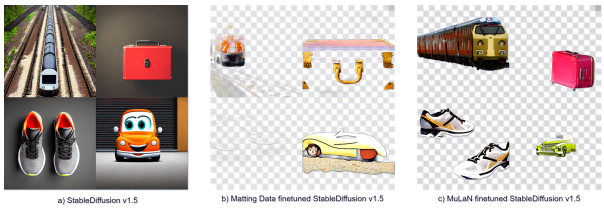
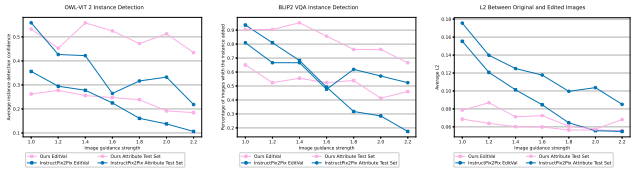


Figure 9. RGBA generation results. Captions: “a train is approaching“, “a red suitcase“, “a pair of running shoes“, “a cartoon car is parked“. For StableDiffusion, “on a black background“ was added.



(a) OwlViT 2 edit detection average confidence (b) BLIP2 VQA edit detection rate (c) L2 between edited and original images

Figure 10. Instance addition. Quantitative analysis.

due to a better understanding of transparency channels.

Instance addition. Our second application considers an image editing task where the objective is to add instances to an image. We finetune an InstructPix2Pix [3] model, taking advantage of our ability to seemlessly add or remove instances in our RGBA stacks. Our training data for InstructPix2Pix comprises triplets $(I_{S\leq i}, I_{S\leq i+1}, C_{S_{i+1}})$ where $C_{S_{i+1}}$ is the instance caption of layer $i+1$ and $I_{S\leq i}$ is the RGB image obtained by flattening the incomplete RGBA stack up to layer i. To assess performance, we use EditVal's instance addition evaluation strategy [2]. We report results on the benchmark introduced in [2] (which adds objects without attributes) and build an additional attribute driven evaluation benchmark. Additional details on both evaluation metrics and our benchmark are available in Supp. Materials. Fig. 10 highlights that our model has a better and more consistent performance across the spectrum, in particular with regards to scene preservation. This is further evidenced in Fig 11, where it can clearly be seen that our model has substantially lower attribute bleeding and better background preservation. This can be attributed to our training setup guaranteeing background preservation, in contrast with InstructPix2Pix using Prompt-to-prompt [12] editing results.

5. Conclusions

In this work, we introduce MuLAn, a novel dataset for generative AI development that comprises over 44K multi-layer annotation of decomposed RGB images. We built MuLAn by processing images from the LAION Aesthetic 6.5 and COCO datasets using a novel pipeline capable of decomposing RGB images in multi-layer RGBA stacks. MuLAn offers a wide range of scene types, image styles, resolutions and object

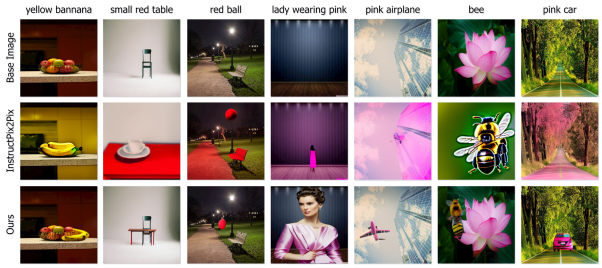


Figure 11. Instance addition. Qualitative examples.

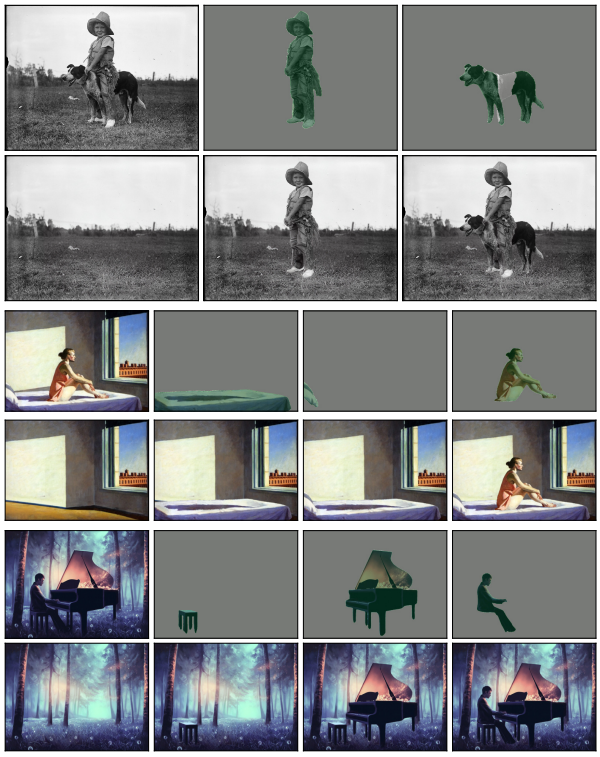


Figure 12. Visualisation of 3 decompositions from MuLAn-COCO (top) and MuLAn-LAION (bottom 2). From left to right: original image, instance RGBA image with green alpha overlay (top row); reconstructed images by adding layers one by one (bottom row).

categories. By releasing MuLAn, we aim to open new possibilities in compositional text-to-image generative research. Key to building MuLAn is our image decomposition pipeline. We have provided a detailed analysis of the pipeline's failure modes, notably segmentation, detection and inpainting. Future work will investigate solutions to improve performance and increase MuLAn's size. We can notably leverage the modular nature of the pipeline to introduce better performing models, *e.g.* segmenters or inpainters. Additionally, the pipeline can be used as a standalone solution to decompose images and facilitate editing with common software. To support this, we additionally investigate human in the loop extensions.

References

[1] Jiayang Ao, Qiuhong Ke, and Krista A Ehinger. Image amodal completion: A survey. *Computer Vision and Image Understanding*, page 103661, 2023. 2, 5

[2] Samyadeep Basu, Mehrdad Saberi, Shweta Bhardwaj, Atoosa Malemir Chegini, Daniela Massiceti, Maziar Sanjabi, Shell Xu Hu, and Soheil Feizi. Editval: Benchmarking diffusion based text-guided image editing methods. *arXiv preprint arXiv:2310.02426*, 2023. 8

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 8

[4] Pierre Buyssens, Maxime Daisy, David Tschumperlé, and Olivier Lézoray. Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions. *IEEE transactions on image processing*, 24(6):1809–1824, 2015. 5

[5] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 2

[6] Helisa Dhamo, Nassir Navab, and Federico Tombari. Object-driven multi-layer scene decomposition from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5369–5378, 2019. 2, 3

[7] Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6144–6153, 2018. 2, 3

[8] Dave Epstein, Allan Jabri, Ben Poole, Alexei A Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. *arXiv preprint arXiv:2306.00986*, 2023. 2

[9] Violeta Menéndez González, Andrew Gilbert, Graeme Phillipson, Stephen Jolly, and Simon Hadfield. Sainet: Stereo aware inpainting behind objects with generative networks. *arXiv preprint arXiv:2205.07014*, 2022. 3

[10] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 7

[11] Martin N Hebart, Adam H Dickter, Alexis Kidder, Wan Y Kwok, Anna Corriveau, Caitlin Van Wicklin, and Chris I Baker. Things: A database of 1,854 object concepts and more than 26,000 naturalistic object images. *PloS one*, 14(10): e0223792, 2019. 4

[12] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2, 8

[13] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander G Schwing. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3105–3115, 2019. 2, 3

[14] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *arXiv preprint arXiv:2307.06350*, 2023. 1

[15] Kyong Hwan Jin and Jong Chul Ye. Annihilating filter-based low-rank hankel matrix approach for image inpainting. *IEEE Transactions on Image Processing*, 24(11):3498–3511, 2015. 5

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 4

[17] Hyunmin Lee and Jaesik Park. Instance-wise occlusion and depth orders in natural scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21210–21221, 2022. 4

[18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 6

[19] Pengzhi Li, Qinxuan Huang, Yikang Ding, and Zhiheng Li. Layerdiffusion: Layered controlled image editing with diffusion models. In *SIGGRAPH Asia 2023 Technical Communications*, pages 1–4. 2023. 2

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 2, 6, 7

[21] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 6

[22] Erika Lu, Forrester Cole, Tali Dekel, Weidi Xie, Andrew Zisserman, David Salesin, William T Freeman, and Michael Rubinstein. Layered neural rendering for retiming people in video. *arXiv preprint arXiv:2009.07833*, 2020. 3

[23] Shilin Lu, Yanzhu Liu, and Adams Wai-Kin Kong. Tf-icon: Diffusion-based training-free cross-domain image composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2294–2305, 2023. 2

[24] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 4

[25] B. E. Moore and J. J. Corso. Fiftyone. *GitHub. Note: https://github.com/voxel51/fiftyone*, 2020. 6

[26] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 2

[27] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[29] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. 4

[30] N Dinesh Reddy, Robert Tamburo, and Srinivasa G Narasimhan. Walt: Watch and learn 2d amodal representation from time-lapse imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9356–9366, 2022. 2, 3

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 7

[32] Vishnu Sarukkai, Linden Li, Arden Ma, Christopher Ré, and Kayvon Fatahalian. Collage diffusion. *arXiv preprint arXiv:2303.00262*, 2023. 2, 3

[33] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2, 6

[34] Samyakh Tukra, Hani J Marcus, and Stamatia Giannarou. See-through vision with unsupervised scene occlusion reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3779–3790, 2021. 3

[35] Xiaosheng Yan, Feigege Wang, Wenxi Liu, Yuanlong Yu, Shengfeng He, and Jia Pan. Visualizing the invisible: Occluded vehicle segmentation and recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7618–7627, 2019. 2, 3

[36] Jingfeng Yao, Xinggang Wang, Lang Ye, and Wenyu Liu. Matte anything: Interactive natural image matting with segment anything models. *arXiv preprint arXiv:2306.04121*, 2023. 6

[37] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *arXiv preprint arXiv:2209.09407*, 2022. 4

[38] Lewei Yao, Jianhua Han, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, and Hang Xu. Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23497–23506, 2023. 4

[39] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3784–3792, 2020. 3

[40] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2

[41] Xinyang Zhang, Wentian Zhao, Xin Lu, and Jeff Chien. Text2layer: Layered image generation using latent diffusion model. *arXiv preprint arXiv:2307.09781*, 2023. 2, 3, 7

[42] Chuanxia Zheng, Duy-Son Dao, Guoxian Song, Tat-Jen Cham, and Jianfei Cai. Visiting the invisible: Layer-by-layer completed scene decomposition. *International Journal of Computer Vision*, 129:3195–3215, 2021. 2, 3

[43] Qiang Zhou, Shiyin Wang, Yitong Wang, Zilong Huang, and Xinggang Wang. Human de-occlusion: Invisible perception and recovery for humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3691–3701, 2021. 2, 3