Supplementary Materials - 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

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

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

Supplementary Material

1. Data Filtering models

In order to filter out failed decompositions, we train two classifiers using our 5000 manually annotated decomposition results. The first classifier takes as input the original image, inpainted background image and background inpainting mask. It is a three-way classifier separating successful decompositions from background inpainting failures and irrelevant decomposition. The second one operates at the instance level, taking as input original image, background inpainting mask, inpainted instance and instance alpha mask; it is designed as a multiclass classifier identifying good decompositions, detection, segmentation and inpainting issues, and truncated instances. Both classifiers are built using a frozen pre-trained EfficientNet B0 backbone [23], with the exception of the first layer which is replaced to handle the different input channel size. The background classifier simply trains a fully connected layer on annotated data using a cross entropy loss.

For our instance level classifier, we adopt a more complex strategy: our annotations are image level, while issues are often encountered at the level of a single instance. Taking inspiration from Multiple Instance Learning (MIL) approaches for weak supervision [21], we design a multilabel MIL classification task. Each decomposed image represents a bag of instances, with a set of image level categories (good, segmentation, detection, inpainting, truncated). For a given category, a label of 1 indicates that *at least one* instance in this image has this label, while 0 indicates that no instance has this label. To train this model, we first compute individual instance representations using our EfficientNet backbone, then compute a joint image representation using a self-attention mechanism across all image instances [21]. We then feed this global feature vector to a learnable multi-label classifier and train the model using an image-level cross entropy loss.

We train both models for 200 epochs with learning rates $2e^{-3}$ (background) and $2e^{-5}$ (instance) for a batch size of 16. Our self attention layer has a single head, and dimension 512, requiring an additional projection layer at the EfficientNet output. For each classifier, we reserve 20% of annotated data for validation purpose. Due to the class imbalance between successful decomposition and rarer failure modes, we adopt a square root sampling strategy [18] to train our background classifier, sampling rare classes more often. Performance of filtering models on the validation set is reported in table S1. We report F1 scores, which are more accurate evaluators of imbalanced classification results.

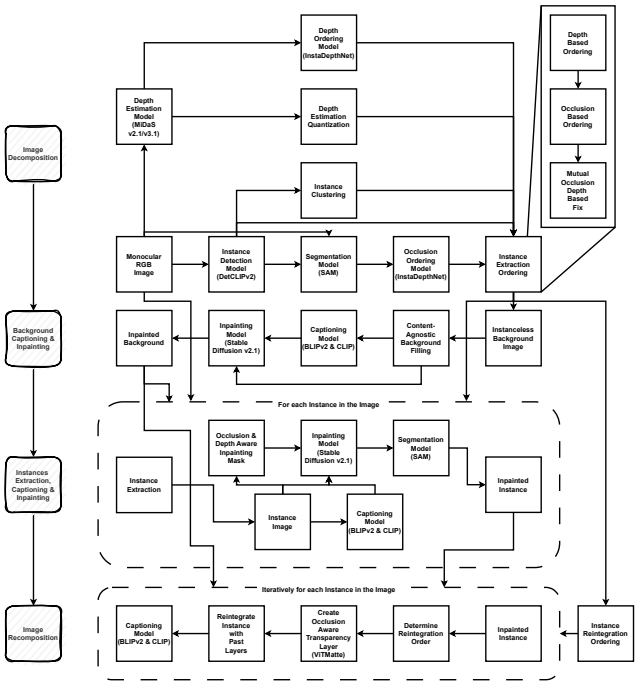

Figure S1. Detailed depiction of the pipeline.

2. Pipeline Embodiment

2.1. Detailed implementation

A detailed schematic of our decomposition pipeline is available in Fig. S1. We provide additional implementation details below.

Detection. A reimplementation of DetCLIP v2 [28] is used with a SWIN-L backbone. We use an instance score threshold of 0.25, and a Non Maximal Suppression threshold of 0.9. Our class list is attached to this Supplementary Material.

Segmentation. We use the Segment Anything VIT-h Model [9] as our segmentation model. We use the DetCLIP v2 bounding box predictions as grounding inputs to the model, and as post-processing, exclude instances whose largest connected component is smaller than 20 pixels or 0.1 % of the whole image. This prevents errors in the matting process (TriMap computation) associated with instance pixel counts being too small

Weights found at on Meta AI Public Files.

Filtering Model	Success label	Reject label			
Full image classifier	0.94	Background inpainting		Irrelevant decomposition	
		0.81		0.81	
Instance classifier	0.96	Segmentation	Inpainting	Truncated instance	Detection
		0.88	0.76		0.90

Table S1. F1 scores measuring performance of data filtering models.

Depth Estimation. MiDaS v3.1 BEiTL-512 [20] is used for its robustness and performance. The depth estimation is quantised in bins of 250 relative depth units in order to increase instance separability for the initial instance extraction. Images are resized to 512×512 for depth estimation precision.

Instance Occlusion Ordering. We used the $InstaDepthNet^{o,d}$ [11] model to predict both an initial depth ordering as well as an occlusion ordering. The model is build on top of MiDaS v2.0 and takes as input the instance segmentations generated by SAM.

Captioning. ViT-g FlanT5XL trained via the BLIP-2 paradigm [13] and further finetuned to produce COCO style captions [14] is used to caption the instances, background and intermediary layers. The captioning prompt that was used is the one used during training *"a photo of"*. In order to increase reproducibility, Beam Search is used with top 32 beams being kept in memory. Besides the 32 predicted captions, the instance's category name and the term "image" for the background layer were added as candidate captions. The best caption was then selected using a pretrained CLIP model with VIT-L-patch14 backbone [19] based on the similarity score between caption and image. Lastly, we used LLaVa v1.5 7b [15] to generate detailed captions of the background and the fully recomposed image in order to promote complex captioning based generation [3]. LLaVa was not used on individual instances as it was hallucinating too many details based on the instance's appearance (*e.g.* person's pose). The captioning component takes as input the instance extracted based on the SAM segmentation.

Inpainting. We use Stable Diffusion v1.5 for our inpainting task. We dilate bounding boxes by a 0.1 ratio of image size, and crop the input image within the dilated bounding box. This cropped image is used as input to the inpainting model. Inpainting masks are dilated as well using a Guassian blur filter with $\sigma = 7$. The difference in sigma is required to guarantee that for the background inpainting instance contents such as hairs are presents. The area within the inpainting mask is filled with a constant value based on the image content .

Inpainting is carried out with 50 timesteps. After inpainting, the cropped area is reintroduced and merged with original image pixels according to the inpainting mask. This reduces content degradation, notably from VAE encoding-decoding. To ensure smooth merging, we dilate inpainting masks and soften mask edges using gaussian blur.

For background inpainting, we use the same prompt across all images: *"an empty scene"* and the following negative prompts: *"complex, text, distortions, poor quality, crowded, non-uniform, item, main subject, large object, foreground object, foreground, heterogeneous, man, woman"*. We additionally append all detected category names in each image to the negative prompts.

For instance inpainting, we use estimated captions as prompt and the following negative prompts: *"complex, text, poor quality, distortions, crowded, bad anatomy, deformed, missing arms, missing hands, missing legs, extra arms, extra legs, NSFW, nsfw, tiling, bad proportions, cropped, unnatural pose, fused fingers, missing fingers"*. We additionally append all detected category names (that do not pertain to the instance of interest) in each image to the negative prompts.

Matting. We use ViT-Matte [26] finetuned on Composition 1K together with SAM, following the MatteAnything approach [27]. We resize the mask predicted by SAM to 256×256, and use the dilated bounding box as grounding. Then the mask is both eroded and dilated with a kernel size of 2 for 2 iterations in order to automatically generate a TriMap. The dilation and erosion are conservative in order to preserve small instances. We then use ViT-Matte to predict the Alpha channel based on the inpainted instance image and the TriMap. All values below 0.1 are set to 0 to delete the sporadic alpha noise. Then a matting mask is generated from the alpha channel by binarising it. This matting mask is used to extract the matted instance from its inpainted representation. We have mainly done this due to the unreliable nature of inpainting.

Weights found on GitHub.
Weights found on HuggingFace.
Weights found on HuggingFace.
Weights found on HuggingFace.

Based on Stable diffusion webui.
Weights found on Google Drive.

2.2. Instance Ordering

Algorithm. 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 further away to closest (according to instance mean depth value). This can easily be achieved using the instance depth graph, by computing node outdegree: 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. In algorithm 1, we provide an algorithmic overview of our instance ordering algorithm, to facilitate reader comprehension.

Quantitative analysis. To assess the performance of our novel ordering strategy, we ablate over its components and use three metrics to evaluate image reconstruction quality: LPIPS [30], SSIM [24] and MSE. LPIPS uses deep features across multiple scales from the AlexNet [10] to measure the similarity between a perturbed image and the ground truth image, and was shown to align well with human evaluation [30]. SSIM is a staple reconstruction metric used across multiple domains. It is the holistic product of three local dissimilarity factors, namely, luminance, variance, and correlation. Finally we used MSE on both the whole image as well the area covered by the background inpainting mask (MSE Masked). All the metrics were calculated only on images with strictly positive bounding box IoUs (*i.e*. with overlapping instances).

We compare our ordering method to four baselines. First, *Reverse Decomposition* is the reverse ordering used to extract instances from the image, as detailed in Section 3.1-Instance Extraction. Other baselines are ablations of our ordering strategy and are equivalent to compound effects of the steps from Algorithm 1. *Depth Ordering* sorts instances based on their mean depth value, *+Occlusion Resolution* further adjusts ordering based on occlusion information, *+Mutual-Occlusion Resolution* integrates order correction based on mutual-occlusion information and constitutes the complete ordering process. All reconstructed images have alpha channels that have been predicted by the ViT Matte model. Finally, we further evaluate the impact of our occlusion aware alpha estimation (*+Occlusion Altered Alpha*), where mutually occluded areas are set to be transparent.

Our depth based ordering achieves the worst reconstruction quality, highlighting the limitations of relying solely on depth information. Second is our reverse decomposition baseline, which relies on clustering and bounding box size heuristics in addition to depth. Replacing these heuristics with occlusion and mutual occlusion based corrections further improve reconstruction quality, with our final ordering approach achieving the top performance. Finally, our occlusion aware alpha estimation further improves image fidelity by a noticeable margin. We additionally provide sub-dataset specific results for our complete approach, showing that MuLAn-LAION generally achieves better reconstruction quality than MuLAn-COCO. We attribute this difference to the higher image complexity of COCO dataset when compared with LAION Aesthetic 6.5 (*e.g*. more single instance scenes in MuLAn-LAION).

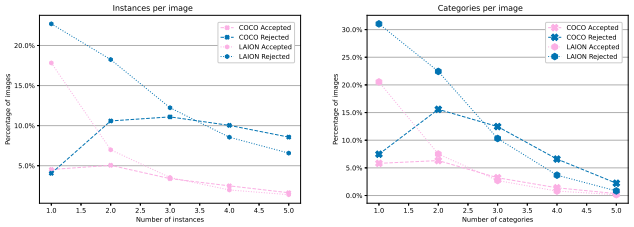

Figure S2. Scene distribution for successful and rejected decompositions, for both COCO and LAION datasets. Percentage of accepted/rejected decompositions with respect to number of instances (left), and number of categories (right) per image.

3. Dataset details

3.1. Usage

MuLAn is split into two subsets based on the original datasets that are annotated. MuLAn-COCO consists of 16,034 images with 40,335 instances while MuLAn-LAION consists of 28,826 images with 60,934 instances, for a sum total of 44,860 images with 101,269 instances.

Additional dataset statistics Fig. S2 shows the percentage of successful and failed decompositions with respect to the number of categories and instances in an image. Differences in scene composition between both datasets are highlighted in this figure: we can see that LAION has a much larger distribution of simpler scenes (1-2 instances and categories), while COCO has a more balanced distribution. In both cases, percentages of successful decompositions decrease as the number of instances increase, highlighting the challenge of handling the intricacies of complex scenes.

The complete distribution of categories in our dataset is available in Fig. S4 (MuLAn-COCO) and Fig. S5 (MuLAn-LAION). These figures additionally highlight which categories have the highest success and failure rates, showing the proportion of accepted and rejected examples.

In addition, we evidence robust performance and an ability to generalise across a wide-range of image resolutions and qualities as shown in Fig. S3, where we report the distribution of image resolutions in our dataset.

Algorithm 1: Instance ordering procedure

Inputs: A non sorted list of instances $\mathcal{N}$,
A list L^D of maximum depth values per instance,
A graph $\mathcal{G}^D = (\mathcal{N}, \mathcal{E}^D)$ of relative depths, where $e^D_{ij} \in \mathcal{E}^D$ if instance i is in front of instance j (lower depth)
A graph $\mathcal{G}^O = (\mathcal{N}, \mathcal{E}^O)$ of relative occlusions, where $e^O_{ij} \in \mathcal{E}^O$ if i occludes j
Output: $\mathcal{N}^S$ sorted in inpainting order

Depth based ordering
  Compute node depth outdegree $\forall n \in \mathcal{N}$: $outdeg(n) = \sum_i |\mathcal{E}^D_{ni}|$ ▷ Number of instances behind n
  $\mathcal{N}^S \leftarrow$ Sort $\mathcal{N}$ by ascending outdegree value

Occlusion based ordering correction
  For $i = 1$ **to** $|\mathcal{N}^S| - 1$: ▷ Loop following current order $\mathcal{N}^S$
    For $j = i + 1$ **to** $|\mathcal{N}^S|$: ▷ Loop through instances inpainted after i
      if $e^O_{ij} \in \mathcal{E}^O$ and $e^O_{ji} \notin \mathcal{E}^O$:
        $\mathcal{N}^S \leftarrow Swap(i, j)$ ▷ Swap instances to inpaint occluded instance first

Final adjustment: mutual occlusion
  For $i = 1$ **to** $|\mathcal{N}^S| - 1$:
    For $j = i + 1$ **to** $|\mathcal{N}^S|$:
      if $e^O_{ij} \in \mathcal{E}^O$ and $e^O_{ji} \in \mathcal{E}^O$: ▷ mutual occlusion
        if $L^D_i < L^D_j$: ▷ instance j is behind i in terms of max depth value
          $\mathcal{N}^S \leftarrow Swap(i, j)$ ▷ Swap instances to inpaint instance that is further away first

Ordering Logic	Masked MAE ↓	MAE ↓	PSNR ↑	SSIM [24] ↑	LPIPS [30] ↓↓
Reverse Decomposition	$0.0173_{\pm 0.0244}$	$0.0157_{\pm 0.0139}$	$75.9795_{\pm 5.2819}$	$0.99975_{\pm 0.00058}$	$0.0720_{\pm 0.0530}$
Depth Based	$0.0177_{\pm 0.0253}$	$0.0158_{\pm 0.0142}$	$75.9391_{\pm 5.3078}$	$0.99974_{\pm 0.00062}$	$0.0723_{\pm 0.0534}$
+ Occlusion Resolution	$0.0169_{\pm 0.0245}$	$0.0156_{\pm 0.0139}$	$76.0262_{\pm 5.2599}$	$0.99976_{\pm 0.00059}$	$0.0713_{\pm 0.0523}$
+ Mutual-Occlusion Resolution	$0.0166_{\pm 0.0244}$	$0.0155_{\pm 0.0138}$	$76.0552_{\pm 5.2406}$	$0.99977_{\pm 0.00058}$	$0.0711_{\pm 0.0519}$
+ Occlusion Altered Alpha (ours)	$\mathbf{0.0156_{\pm 0.0229}}$	$\mathbf{0.0152_{\pm 0.0133}}$	$\mathbf{76.1859_{\pm 5.1706}}$	$\mathbf{0.99978_{\pm 0.00054}}$	$\mathbf{0.0700_{\pm 0.0507}}$
MuLAn - COCO	$0.0164_{\pm 0.0175}$	$0.0221_{\pm 0.0161}$	$73.9121_{\pm 4.9423}$	$0.99969_{\pm 0.00068}$	$0.0881_{\pm 0.0525}$
MuLAn - LAION	$0.0121_{\pm 0.0133}$	$0.0115_{\pm 0.0079}$	$77.3296_{\pm 4.4036}$	$0.99983_{\pm 0.00019}$	$0.0665_{\pm 0.0494}$

Table S2. Evaluation of the instance ordering re-composition on 4400 LAION images. Masked MAE is MAE applied only on the recomposed image region that was inpainted based on the inpainting mask of the background.

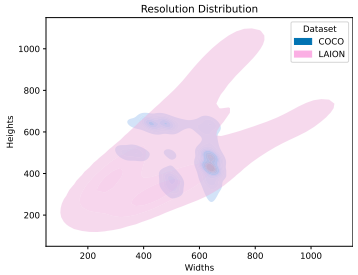

Figure S3. Distribution of image resolutions in MuLAn-COCO and MuLAn-LAION.

3.2. Format

The annotation files are inspired by the original COCO dataset annotations, developed by the research community. The annotation files contain a dictionary with metadata and required elements to generate the MuLAn dataset given the original images and the annotation file. We highlight that we do not release original image content, and that the decomposed images cannot be reconstructed without access to the original data. The dictionary's contents are outlined in Listing 1.

Layers Layers are indexed from 0 to N with Layer 0 being the background. For all layers we have released the masks required to extract the original content from the original

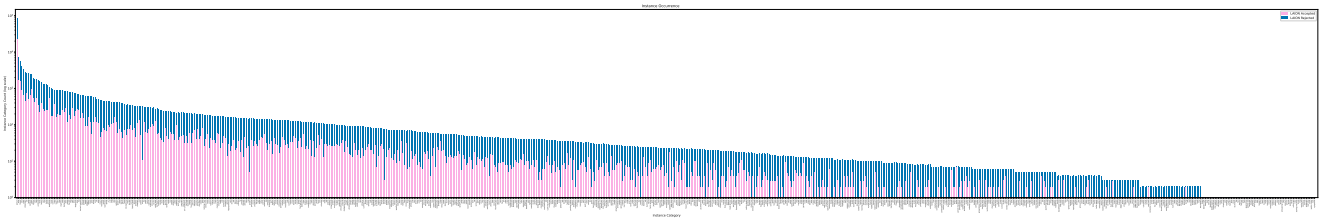

Figure S4. Distribution of all instance categories in the MuLAn-COCO dataset.

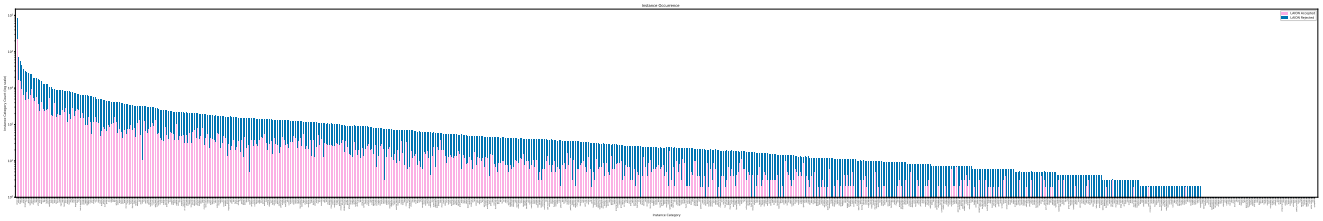

Figure S5. Distribution of all instance categories in the MuLAn-LAION dataset.

image and the inpainted content that needs to be added to the extracted one in order to obtain the layer.

Captions Each individual layer comes with a COCO style caption generated by the FLAN T5-XL BLIP 2 style trained model that was further finetuned on COCO styles [13]. We note that those captions were selected by the CLIP model [4] from 32 candidate captions together with "image" or instance tag as default candidates.

For the background (Layer 0), the original image and the recomposed image we have also released captions generated by the LLaVa model in order to encourage the development of generative models based on detailed natural language captions [3]. The instances were not captioned with LLaVa due to its ability to infer details that were not visible in the instance itself but could be found in the original image. We attribute this to the bias resulting from the position of the instance and the gaps of cotent resulted from isolating the instane to be captioned.

4. Dataset applications details

RGBA Image generation. The matting datasets that have been used in finetuning the baseline SD v1.5 are outlined in Table S3. We make use of 7 publicly available matting datasets for a total 15,791 images (*vs*. 101,269 in MuLAn).

Instance addition. In order to assess the presence of the new instance we used OWL-ViT 2 [17] similarly to the strategy proposed by EditVal [1] for open-set instance detection, and BLIP-2 [13] for visual question answering. For OWL-ViT 2 we report average detection confidence of the detection (in contrast with EditVal's binary scores). For BLIP-2, we report the percentage of images where the model's answer starts with "yes" to the following prompt: *"Question: Answer with yes or no, is there a [*instance description*] in the image? Answer:"*. Following the InstructPix2Pix [2] evaluation, we keep the text guidance scale constant at 7.5 and vary the image guidance scale between 1.0 and 2.2. Since the EditVal instance addition edits do not include attributes we created a small dataset of X edits following their example where some of the edit prompts have attributes. This dataset is the Attribute Test Set and it can be found on the project website .

Dataset	Type	Resolution	No. Instances
AIM-500 [12]	Object	1397×1260	500
AM-2K [12]	Animal	1471×1195	484
HIM-2K [22]	Human	1823×1424	830
RWP-636 [29]	Human	1038×1327	636
PPM-100 [8]	Human	2997×2875	100
Composition 1K [25]	Varied	Varied	481
UGD-12K [5]	Human	357×317	12760

Table S3. Matting datasets used to train our RGBA generation baseline.

5. Choice of dataset

We chose to develop MuLAn based on LAION and COCO datasets due to their pervasiveness within both the generative modelling and computer vision communities. The LAION Aesthetic 6.5 subset was specifically chosen due to an appealing compromise between cardinality, instance density, scene style & content, and image quality. Due to ethical concerns around fair-use of copyrighted content, we do not

Weights found on https://MuLAn-dataset.github.io/

Listing 1. Description of our released annotations for a given decomposed image.

```
"annotation" : {
    "captioning": {
        "llava": LLaVa model details
        "blip2": BLIP 2 model details
        "clip": CLIP model details
    }
    "background": {
        "llava": Detailed background LLaVa caption
        "blip2": COCO style BLIP 2 background caption chosen by CLIP
        "original_image_mask": Original image background content mask
        "inpainted_delta": Additive inpainted background content
    }
    "image": {
        "llava": Detailed original image LLaVa caption
        "blip2": COCO style BLIP 2 original image  caption chosen by CLIP.
    }
    "instances": {
        "blip2": COCO style BLIP 2 instance caption chosen by CLIP.
        "original_image_mask": Original image instance content mask
        "inpainted_delta": Additive inpainted instance content
        "instance_alpha": Alpha layer of the inpainted instance
    }
}
```

include content from original images, and release only our results in annotation format, similar to COCO-based datasets. As such, our data cannot be reconstructed without rightful access to the original image content.

6. What did not work and *why*

In this section we exhaustively enumerate alternative approaches that were explored during our development and the reasons we have chosen the current implementation over the discussed alternatives.

Improvements to the SAM model: SAM-HQ and SEEM We investigated alternatives to the SAM model, which is our main cause of decomposition failures. We first considered SAM-HQ [7], a model advertised as a global improvement to SAM, notably capable of segmenting details (*e.g.* elongated object, wires) more accurately. While we did observed improvements and more precise segmentations for this type of objects, we observed that SAM-HQ also had a higher tendency to oversegment, leading to potentially reduced instance extraction accuracy. We additionally considered the SEEM model [31], which was particularly attractive due to its ability to ground the segmentation using category names. The model was however trained on COCO classes, and we observed reduced performance, compared to SAM, for other categories.

Grounded-SAM. We initially considered Grounded-SAM to extract and segment instances. Grounded-SAM combines Grounding-Dino [16], SAM [9] and Tag2Text [6] in order to predict the instance tags, their bounding box and segmentations. The sequence of three models (vs. two in our final pipeline version) introduced additional unstability. Notably, we used Tag2Text, an image tagger and caption predictor, to identify instances in the images, and provided this list of tags to the Grounding DINO detection model. We observed however that Tag2Text's performance, while very high, was not accurate enough to optimise Grounding DINO's detection performance. In contrast to DETCLIP, Grounding DINO requires the exact list of instances present in the image, and wrong tags can lead to detection of non-existent instances. Ultimately, relying on a single model to detect and identify instances in an image yielded a more robust and consistent performance.

7. Failure modes: visual examples

In Figures S6-S12, we provide several visual examples for all failure modes listed in Sec. 4 of the main paper: object detection (Fig. S6), segmentation (Fig. S11), background (Fig. S10) and instance inpainting (Fig. S8), irrelevant

decomposition (Fig. S12). Object detection can comprise missing instances or double detections. The first affects instance and background inpainting, as missed instances are not included in inpainting masks. The second decomposes instances into two or more subcomponents, which often leads to segmentation artefacts. Segmentation limitations can be linked to the SAM model (under or over-segmentation, segmentation of the wrong area within the input bounding box, checkerboard artefacts) or the VIT-Matte model used for alpha layer generation, typically involving over-segmentation. Background issues typically involve introduction of novel instances or structures inconsistent with the overall scene, and text inpainting, despite the use of dedicated negative prompts. This issue often arises when segmentation is imperfect, leaving small artifacts influencing the inpainting process. Similarly, imperfect segmentation and shape priors can influence instance inpainting, leading to failed reconstruction of occluded areas.

We additionally illustrate limitations of our decomposition pipeline: background occlusions (Fig. S9) and bounding box constrained inpainting (Fig. S7). The former occurs because we treat the background as a single flat layer at the bottom of the RGBA stack, while instances can be occluded by background elements (*e.g.* tree branches, large structures). The latter is a limitation of the SAM model, which requires to input local information on where the instance to segment is. We use a dilated version of the object bounding box as input, as leveraging the inpainting mask can lead to segmentation of irrelevant instances.

8. Additional Decomposition Results

Finally, in Figures S13-S16, we show additional visual results of RGBA decompositions in our MuLAn-LAION and MuLAn-COCO datasets. We highlight the varied scenes, image styles and categories.

9. Additional Dataset Application Results

We provide additional visual results for our dataset application experiments. In Fig. S17, we report qualitative examples of our instance addition experiment, compared to the InstructPix2Pix baseline on our attribute dataset. We can see that we are able to consistently add the desired instance, while at the same time preventing attribute leakage and guaranteeing content preservation.

Fig. S18 provides additional results for our RGBA generator, compared to our Stable Diffusion baselines (original model and model fine-tuned on matting datasets). Our dataset's diversity allows a better prompt understanding and generation ability, with a better grasp of transparency. Notably, we note that our model trained on matting data tends to include backgrounds in generated instances, and to set black pixels in instances as transparent.

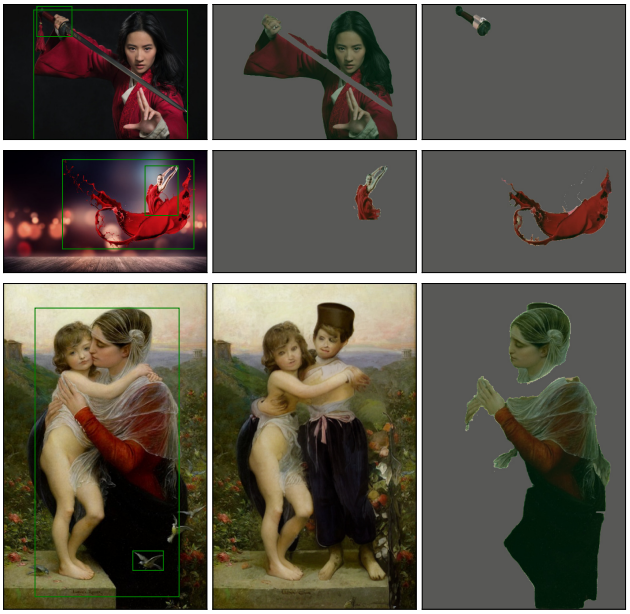

Figure S6. Visualisation of failure modes: object detection. We show detected instance bounding boxes and affected instances/background image.

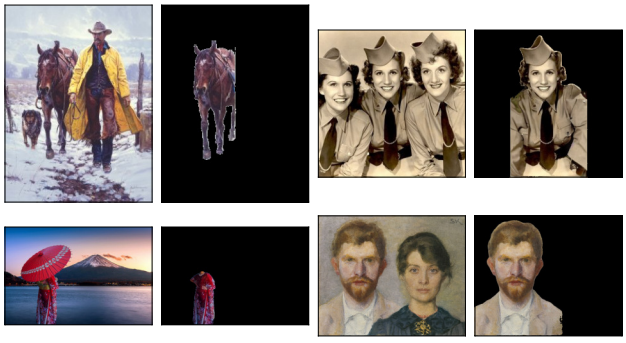

Figure S7. Visualisation of failure modes: bounding box restricted instance completion. Left-right image pairs: original image-instance with failure.

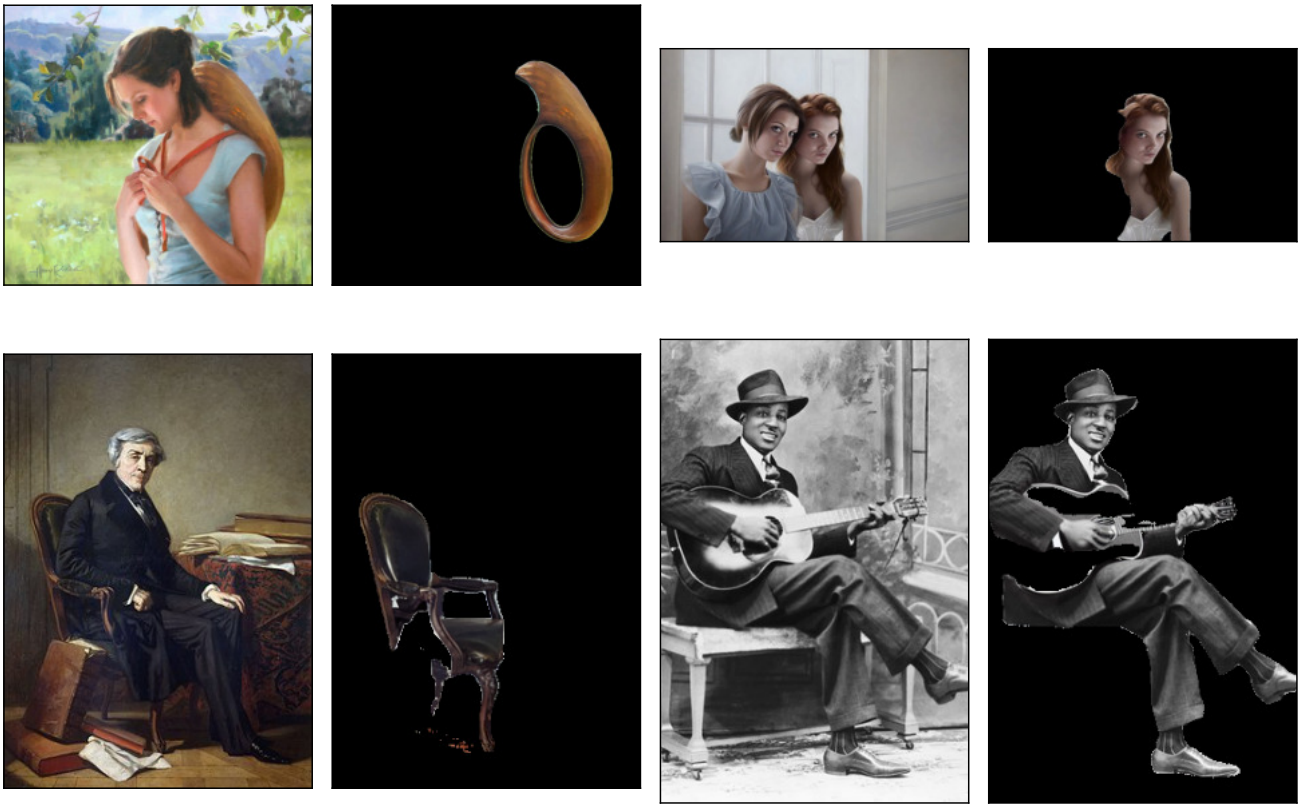

Figure S8. Visualisation of failure modes: instance inpainting. Left-right image pairs: original image-instance with failed inpainting.

Figure S9. Visualisation of failure modes: background occlusions. Green overlay is the estimated instance segmentation.

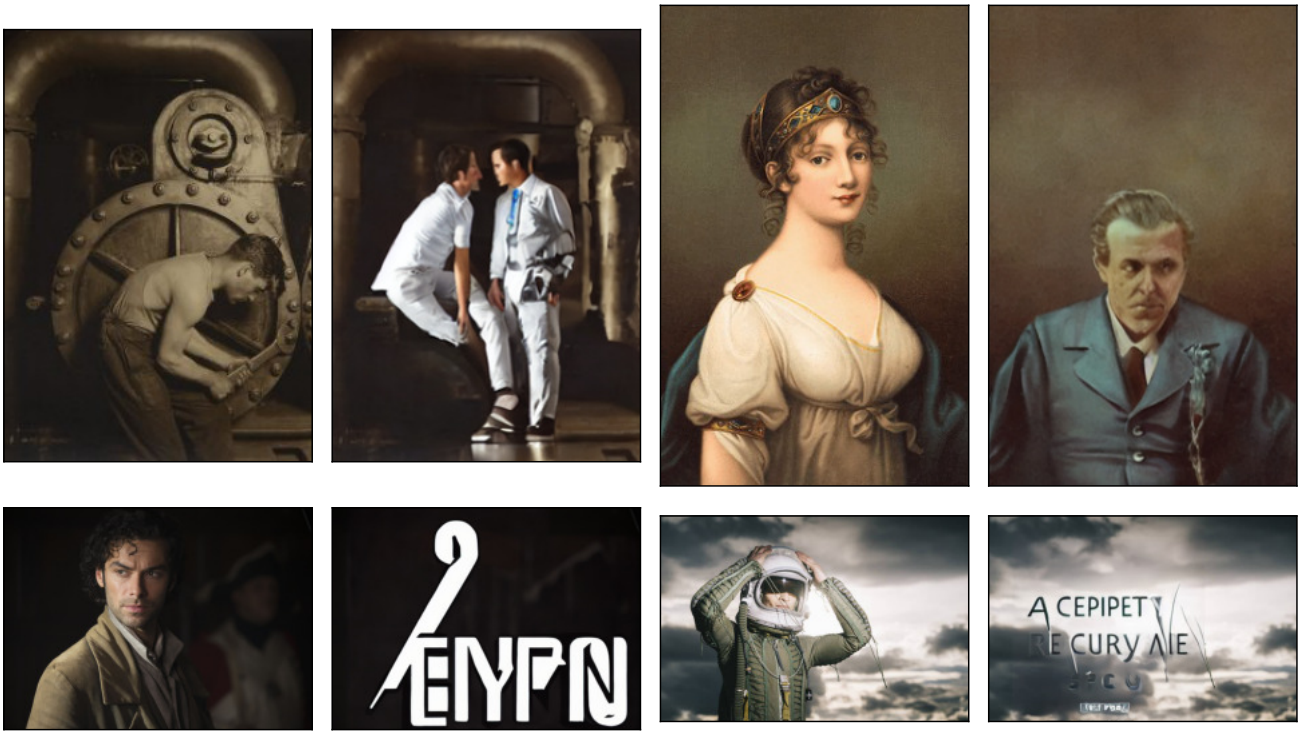

Figure S10. Visualisation of failure modes: background inpainting. Left-right image pairs: original image-inpainted background

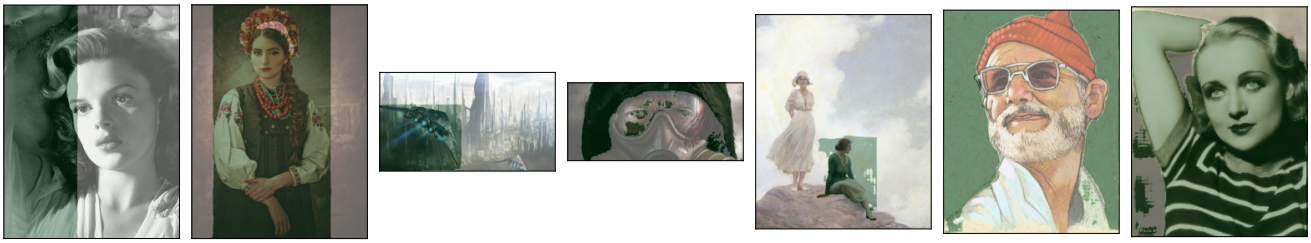

Figure S11. Visualisation of failure modes: segmentation. Green overlay is the estimated instance segmentation.

Figure S12. Visualisation of failure modes: irrelevant decomposition. Bounding boxees show detected objects in the image.

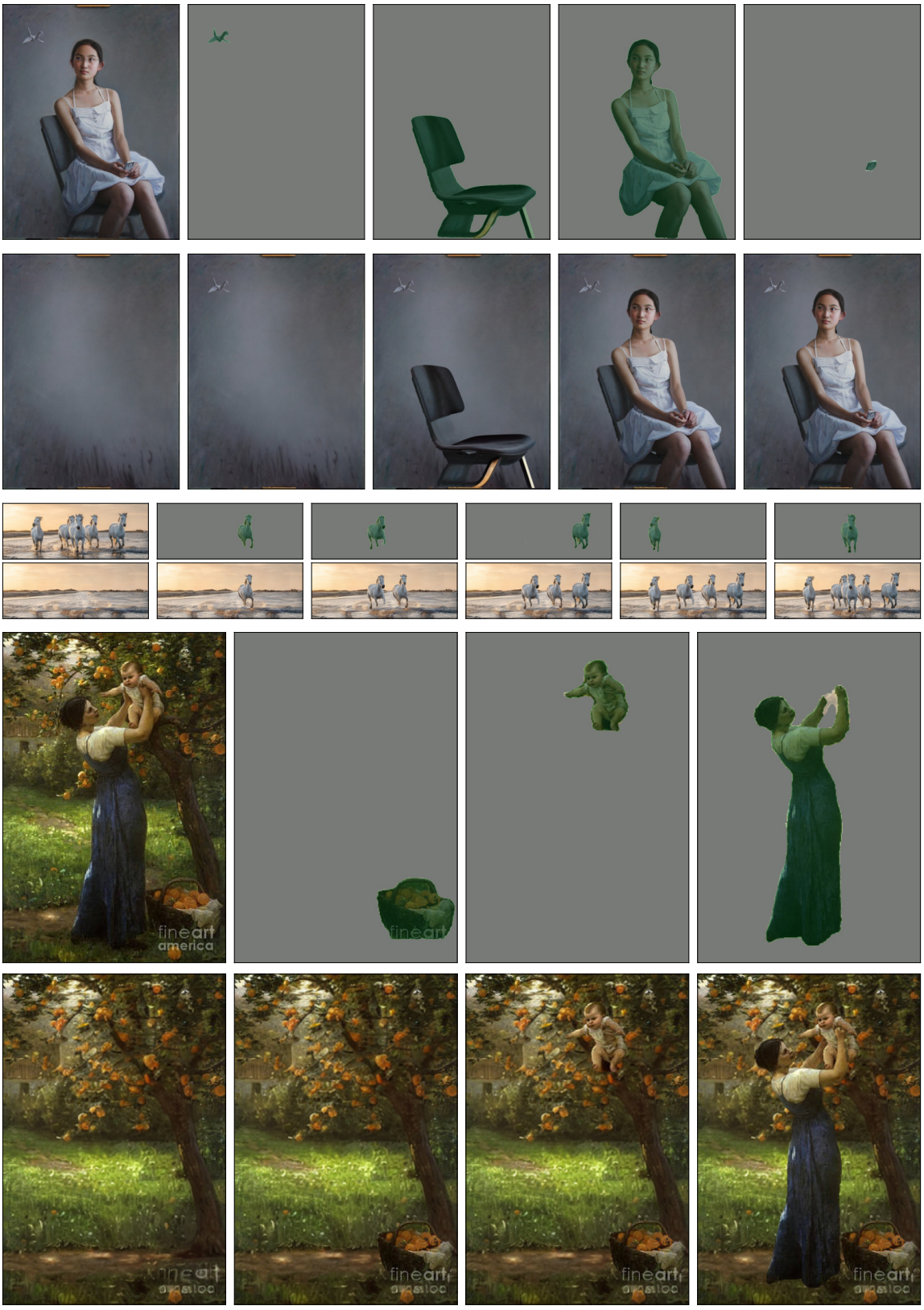

Figure S13. Visualisation of decomposed images from MuLAn-LAION. For each image, from left to right: original image, instance RGBA image with green alpha overlay (top row); progressively reconstructed image by adding layer one by one (bottom row).

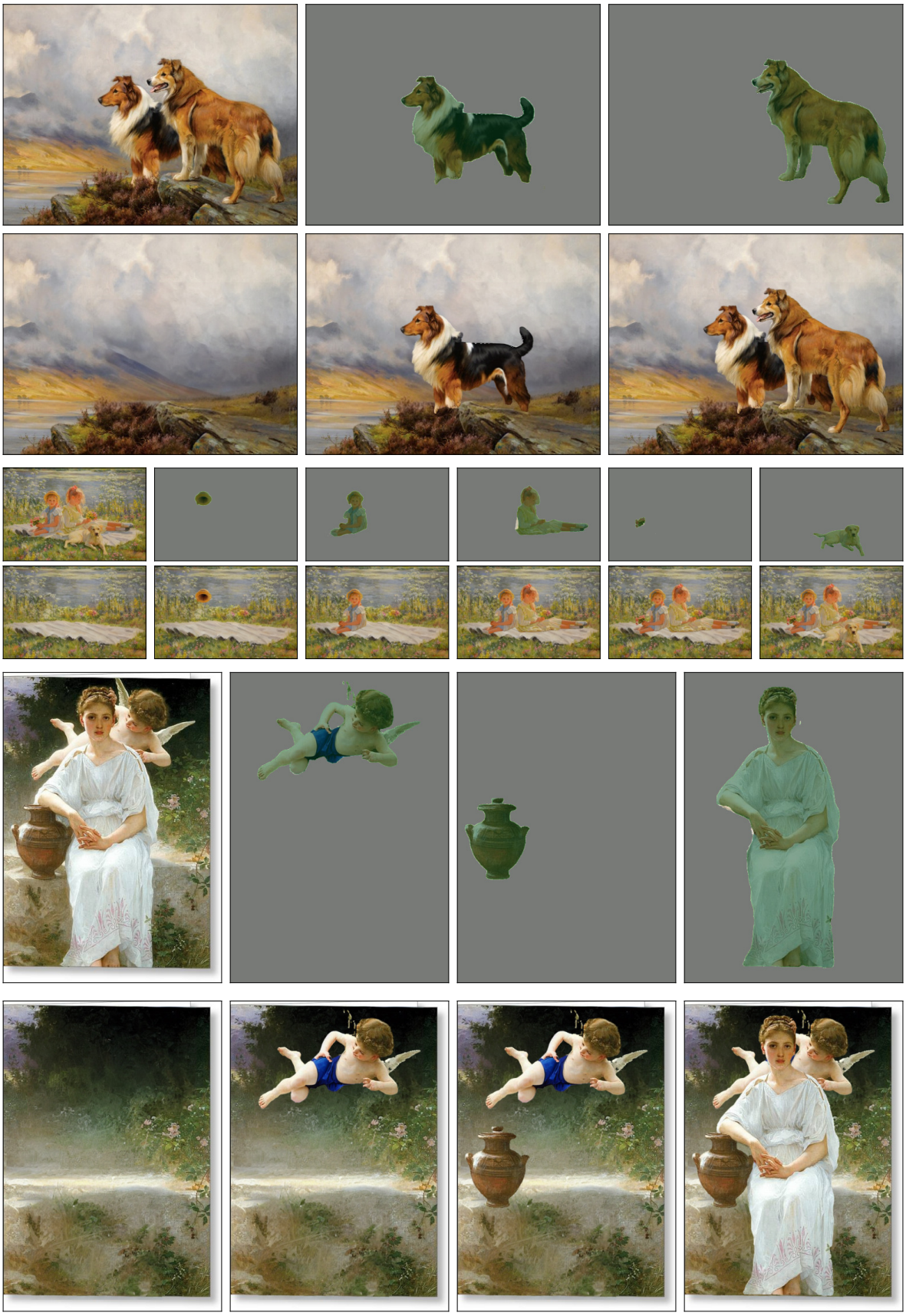

Figure S14. Visualisation of decomposed images from MuLAn-LAION. For each image, from left to right: original image, instance RGBA image with green alpha overlay (top row); progressively reconstructed image by adding layer one by one (bottom row).

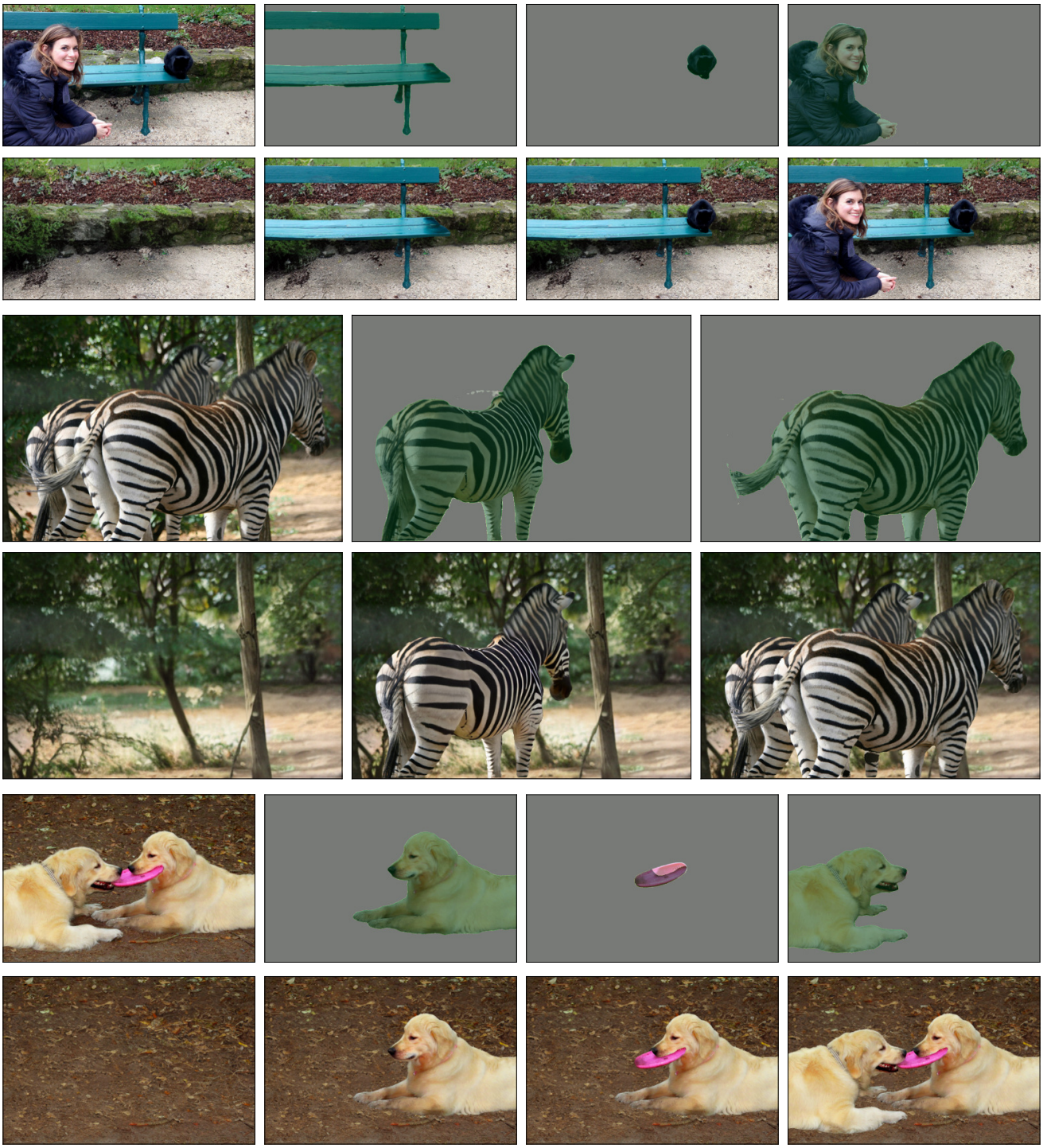

Figure S15. Visualisation of decomposed images from MuLAn-COCO. For each image, from left to right: original image, instance RGBA image with green alpha overlay (top row); progressively reconstructed image by adding layer one by one (bottom row).

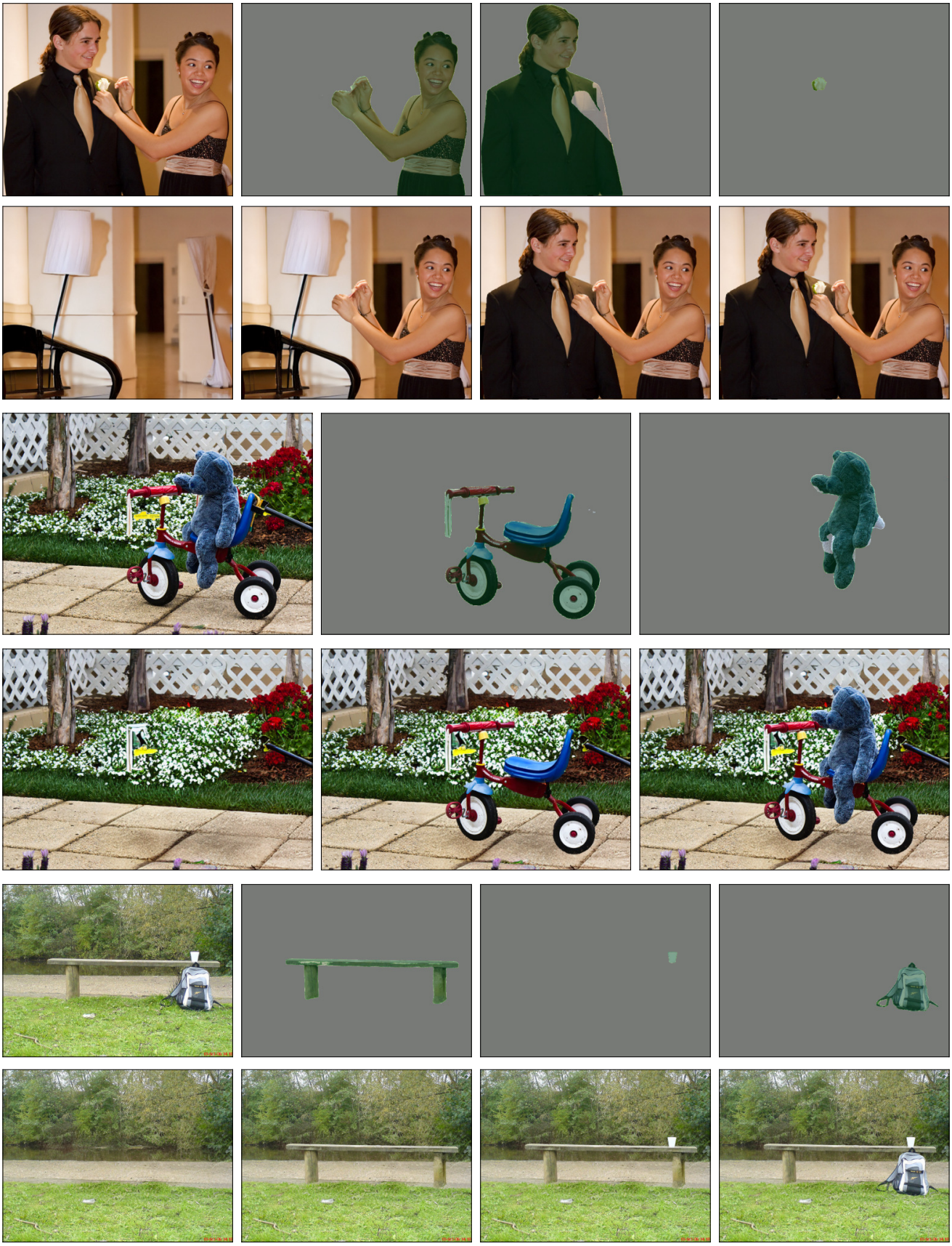

Figure S16. Visualisation of decomposed images from MuLAn-COCO. For each image, from left to right: original image, instance RGBA image with green alpha overlay (top row); progressively reconstructed image by adding layer one by one (bottom row).

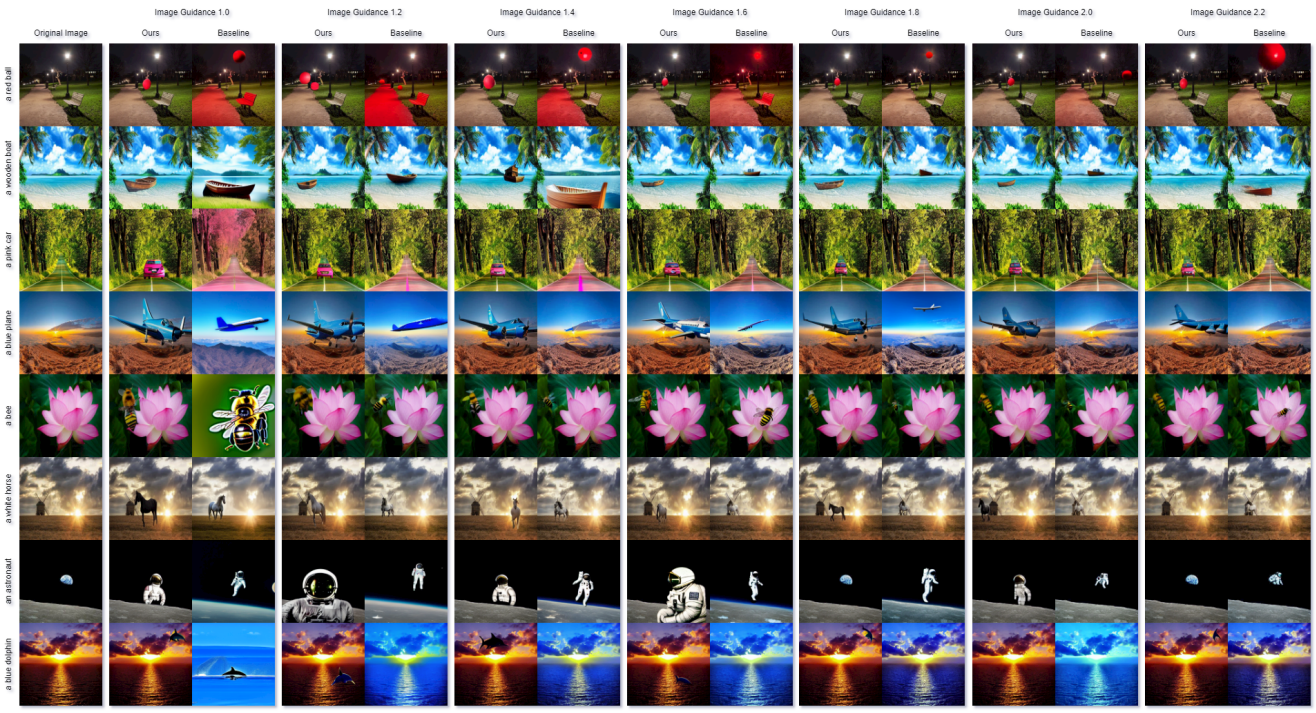

Figure S17. Additional qualitative results of Instance Addition.

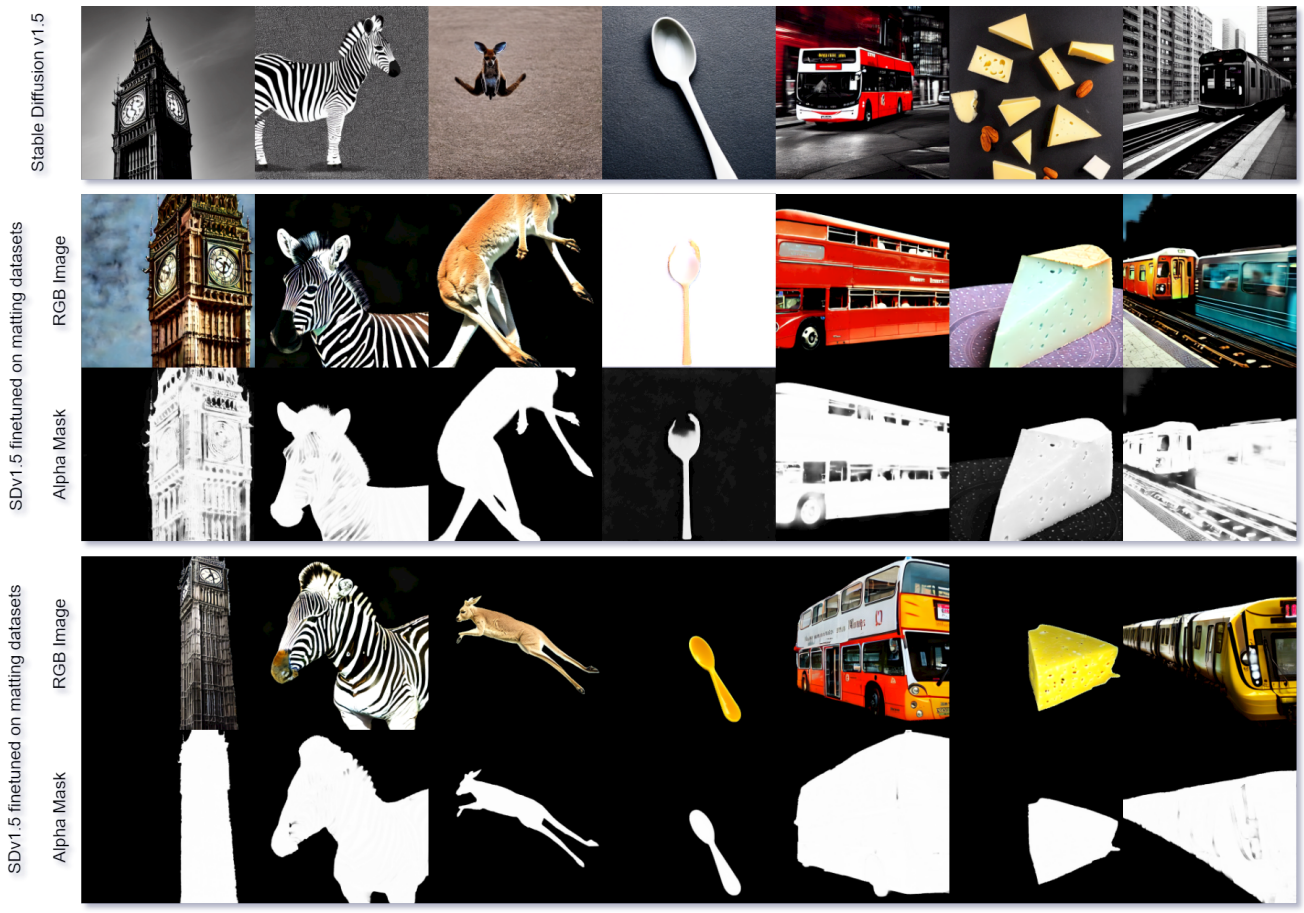

Figure S18. Additional qualitative results of RGBA Generation.

References

[1] 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. 5

[2] 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. 5

[3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 2, 5

[4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. 5

[5] Xiaonan Fang, Song-Hai Zhang, Tao Chen, Xian Wu, Ariel Shamir, and Shi-Min Hu. User-guided deep human image matting using arbitrary trimaps. *IEEE Transactions on Image Processing*, 31:2040–2052, 2022. 5

[6] Xinyu Huang, Youcai Zhang, Jinyu Ma, Weiwei Tian, Rui Feng, Yuejie Zhang, Yaqian Li, Yandong Guo, and Lei Zhang. Tag2text: Guiding vision-language model via image tagging. *arXiv preprint arXiv:2303.05657*, 2023. 6

[7] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *NeurIPS*, 2023. 6

[8] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson W.H. Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *AAAI*, 2022. 5

[9] 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. 1, 6

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 3

[11] 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. 2

[12] Jizhizi Li, Jing Zhang, and Dacheng Tao. Deep automatic natural image matting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 800–806. International Joint Conferences on Artificial Intelligence Organization, 2021. Main Track. 5

[13] 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. 2, 5

[14] 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

[15] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 2

[16] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 6

[17] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers, 2022. 5

[18] Sarah Parisot, Pedro M Esperança, Steven McDonagh, Tamas J Madarasz, Yongxin Yang, and Zhenguo Li. Long-tail recognition via compositional knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6939–6948, 2022. 1

[19] 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. 2

[20] 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. 2

[21] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in Neural Information Processing Systems*, 34:2136–2147, 2021. 1

[22] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Human instance matting via mutual guidance and multi-instance refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 5

[23] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1

[24] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 3, 4

[25] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2970–2979, 2017. 5

[26] Jingfeng Yao, Xinggang Wang, Shusheng Yang, and Baoyuan Wang. Vitmatte: Boosting image matting with pretrained plain vision transformers. *arXiv preprint arXiv:2305.15272*, 2023. 2

[27] 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. 2

[28] 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. 1

[29] Qihang Yu, Jianming Zhang, He Zhang, Yilin Wang, Zhe Lin, Ning Xu, Yutong Bai, and Alan Yuille. Mask guided matting via progressive refinement network. *arXiv preprint arXiv:2012.06722*, 2020. 5

[30] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 3, 4

[31] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *NeurIPS*, 2023. 6