SmartMask: Context Aware High-Fidelity Mask Generation for Fine-grained Object Insertion and Layout Control

Supplementary Material

A. Additional Results

In this section, we include some additional results which could not be included due to space constraints in the main paper. In particular, we demonstrate the ability of our approach to perform context-aware multi-object insertion in Sec. A.1. We also show additional results using *SmartMask* for object insertion with better background preservation in Sec. A.2. We provide additional results analysing the ability of our model to perform mask-free insertion in Sec. A.3. Finally, we include additional results comparing our output mask quality with SmartBrush mask-head outputs [23] and various Inpaint+HQSAM methods in Sec. A.4.

A.1. SmartMask for Multiple Object Insertion

While adding a single object to an input image is useful, in practical applications users would typically want to add multiple objects to the input image in order to obtain a desired output scene. For instance, given an image depicting a grassy field, the user may wish to add multiple objects {*bench, man, woman, dog*} such that the final scene aligns with the context *'a couple with their dog sitting on a bench in a grassy field'* (Fig. 1). In this section, we show that unlike prior works which are limited to adding each object independently, the proposed approach allows the user to perform multiple-object insertion in a context-aware manner.

Results are shown in Fig. 1, 2. In particular, we show comparisons with prior inpainting methods when *a)* all objects (*e.g.* bench, man, woman, dog in Fig. 1) are inserted all at once, and *b)* different objects are inserted in a sequential manner. We observe that when inserting all objects at once, prior works typically lead to 1) incorrect/missing objects (*e.g.* missing dog, additional person in Fig. 1), or, 2) introduce visual-artifacts (*e.g.*, *people facing bench's back*). On the other hand, when adding different objects in sequential manner, we observe that prior works often lead to inconsistency-artifacts. For instance, when adding the {*bench, man, woman, dog*} in Fig. 1, we observe that SD-Inpaint [19] leads to outputs which put the *woman and dog on back of the bench*. Commercial state-of-the-art Adobe GenFill [1] performs better however the *dog and woman do not appear to be sitting on the same bench as man*. Furthermore, the generated objects (*man, woman and dog* in row-5) can appear non-interacting when generated in a sequential manner. *SmartMask* helps address this by allowing the user to first add a coherent sequence of context-aware object masks, before using SDXL-based-ControlNet-Inpaint [22, 24] model to perform precise object insertion.

A similar observation is also found in in Fig. 2, where we see that prior inpainting methods [1, 19] either lead to (a) incorrect objects (*e.g.*, missing child, additional person) when adding all objects at once, or, (b) introduce artifacts (*e.g. woman* in row-3, *man's face* and *woman's dress* in row-5) when adding objects sequentially. Furthermore, the object insertion is done in a context-unaware manner. For instance, when adding *flowers* near the hand of the child in row-5, we observe that Adobe GenFill [1] simply adds a big-flower around the hand region. In contrast, the context-aware ability of *SmartMask* model allows it to add the flowers as a bouquet which is held by the child, thereby allowing more coherent multi-object insertion over prior works.

A.2. Additional Results for Single Object Insertion

In this section, we include additional results on using *SmartMask* for single-object insertion across diverse object categories. Results are shown in Fig. 3. Similar to the results in the main paper, we find that prior image inpainting methods typically lead to huge changes in the image background when adding new objects. For instance, when adding *child to a birthday party scene* (row-1, Fig. 3), we observe that prior works completely remove the background table with birthday decorations. In contrast, the proposed approach is able to generate scene-aware masks which can interact with already existing objects in the input image. This allows it to place the child as if sitting on the table in the original image.

We also observe that the effect of compromised backgroud preservation is even more severe when adding new objects near already existing humans in the input image. For instance, in row-4 (Fig. 3) we observe that when adding a *woman* next to a man sitting on top of mountain rock, prior inpainting works *e.g.*, SDInpaint, SDXL-Inpaint[16, 19] either convert the man himself to a woman, or, add two woman on the image which replace the man altogether. This is clearly undesirable if the user simply wants a final scene with a couple sitting on top of a mountain rock. *SmartMask* helps address this by first predicting a precise object mask for adding the woman on the side of the man, before then using ControlNet-Inpaint [22, 24] model for precise object insertion without affecting the image regions for the man.

A.3. Mask-free Object Insertion

A key advantage of *SmartMask* is that unlike prior works which rely on a user-provided coarse mask, the proposed approach can also be used without user-mask guidance. This allows *SmartMask* to facilitate mask-free object inser-

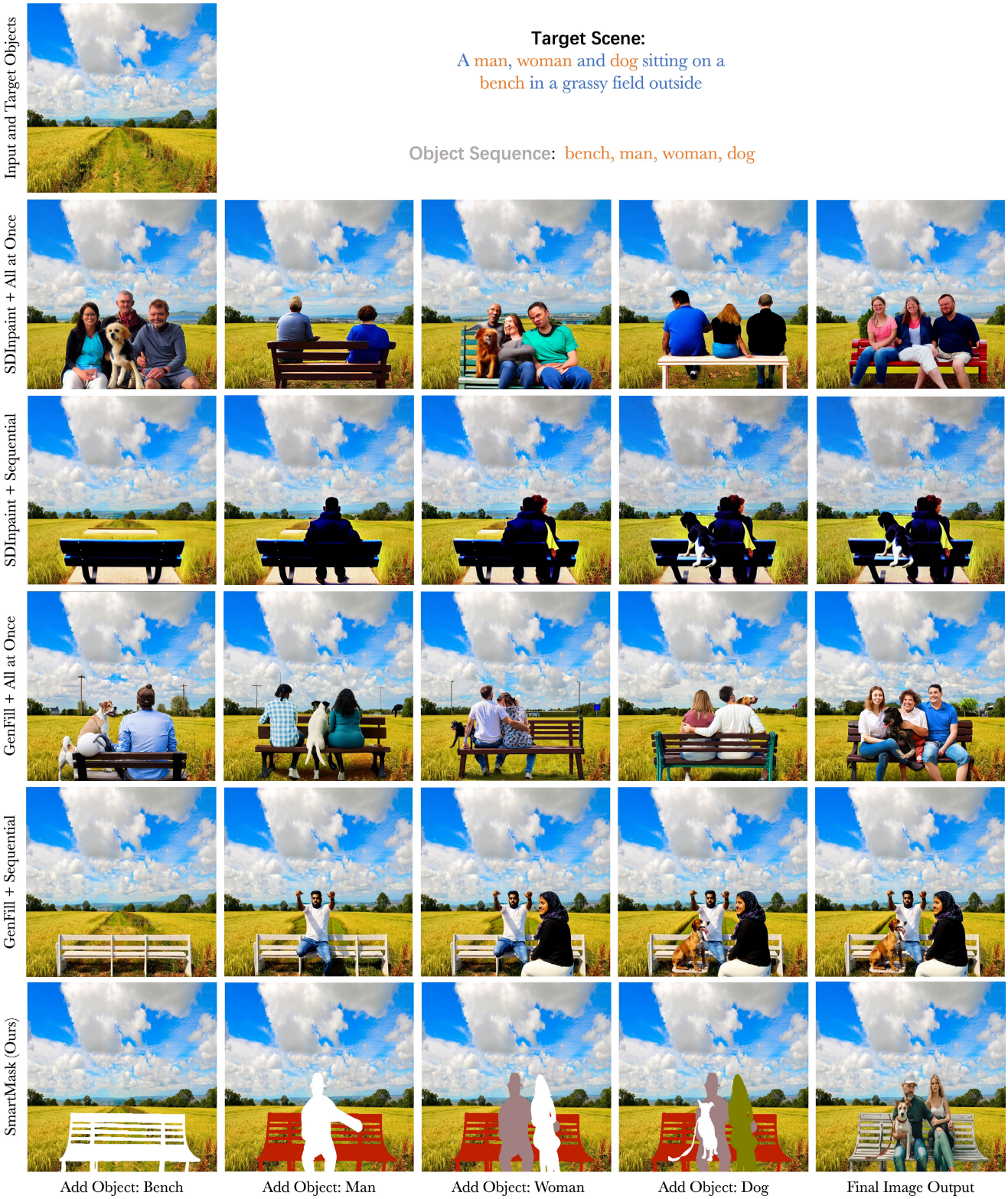

Figure 1. ***SmartMask for multi-object insertion.*** We observe that prior state-of-the-art image inpainting methods [1, 19] either lead to 1) incorrect objects (*missing dog*) or visual-artifacts (*e.g.*, *people facing bench's back*) when adding all objects at once (row-2, row-4), or, 2) introduce inconsistency-artifacts (*e.g.*, *woman and dog in front of bench* in row-3, *dog and woman not sitting on same bench* in row-5) when adding objects sequentially. Furthermore, the generated objects (*man, woman and dog* in row-5) can appear non-interacting when generated in a sequential manner. *SmartMask* helps address this by allowing the user to first add a coherent sequence of context-aware object masks, before using SDXL-based-ControlNet-Inpaint [22, 24] model to perform precise object insertion for multiple objects.

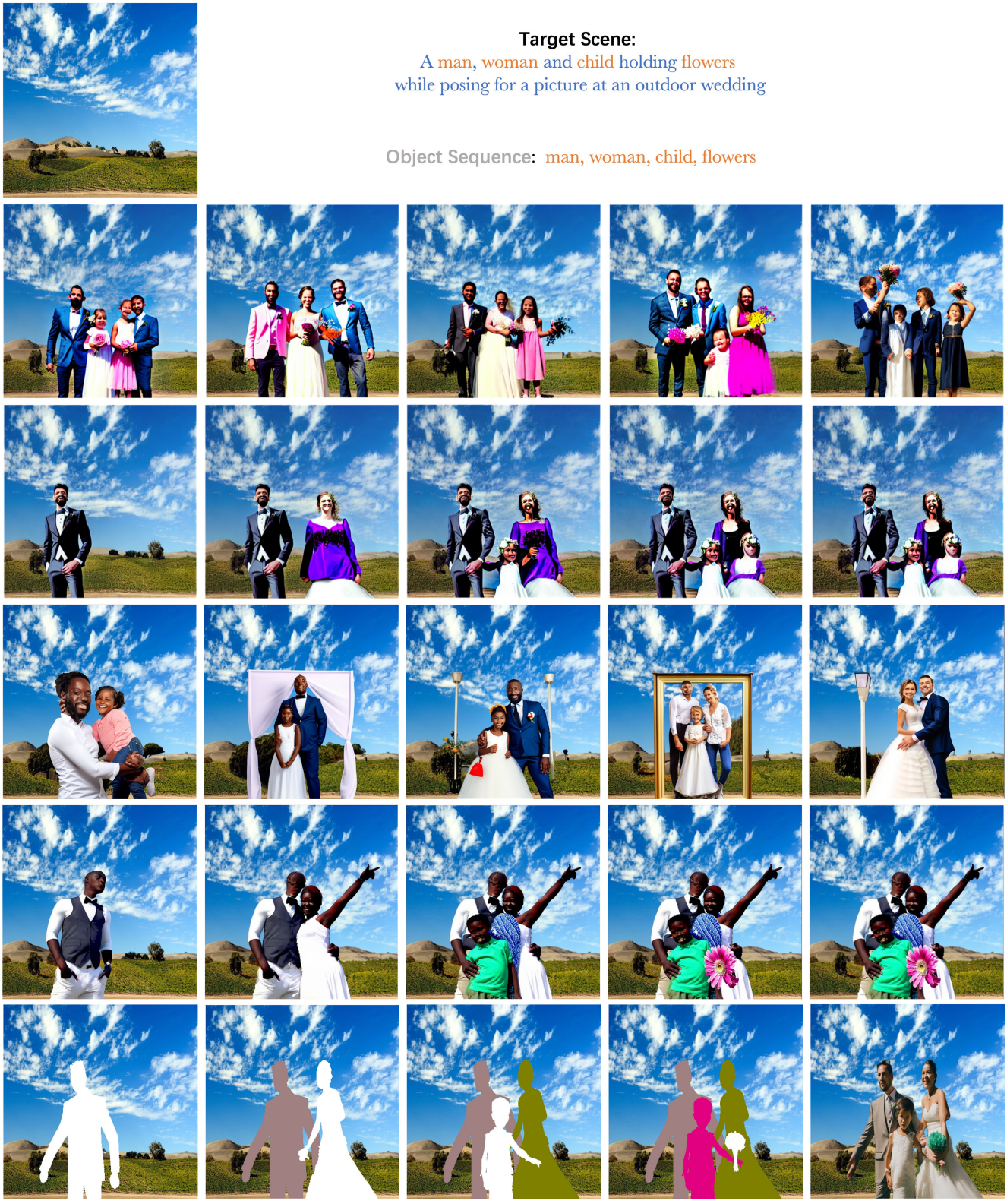

Figure 2. ***SmartMask for multi-object insertion.*** We observe that prior state-of-the-art image inpainting methods [1, 19] either lead to incorrect objects (row-2, row-4) when adding all objects at once, or, introduce artifacts (*e.g.* *woman* in row-3, *man's face* and *woman's dress* in row-5) when adding objects sequentially. *SmartMask* helps address this by allowing the user to first add a sequence of context-aware object masks, before using SDXL-based-ControlNet-Inpaint [22, 24] model to perform precise object insertion for multiple objects.

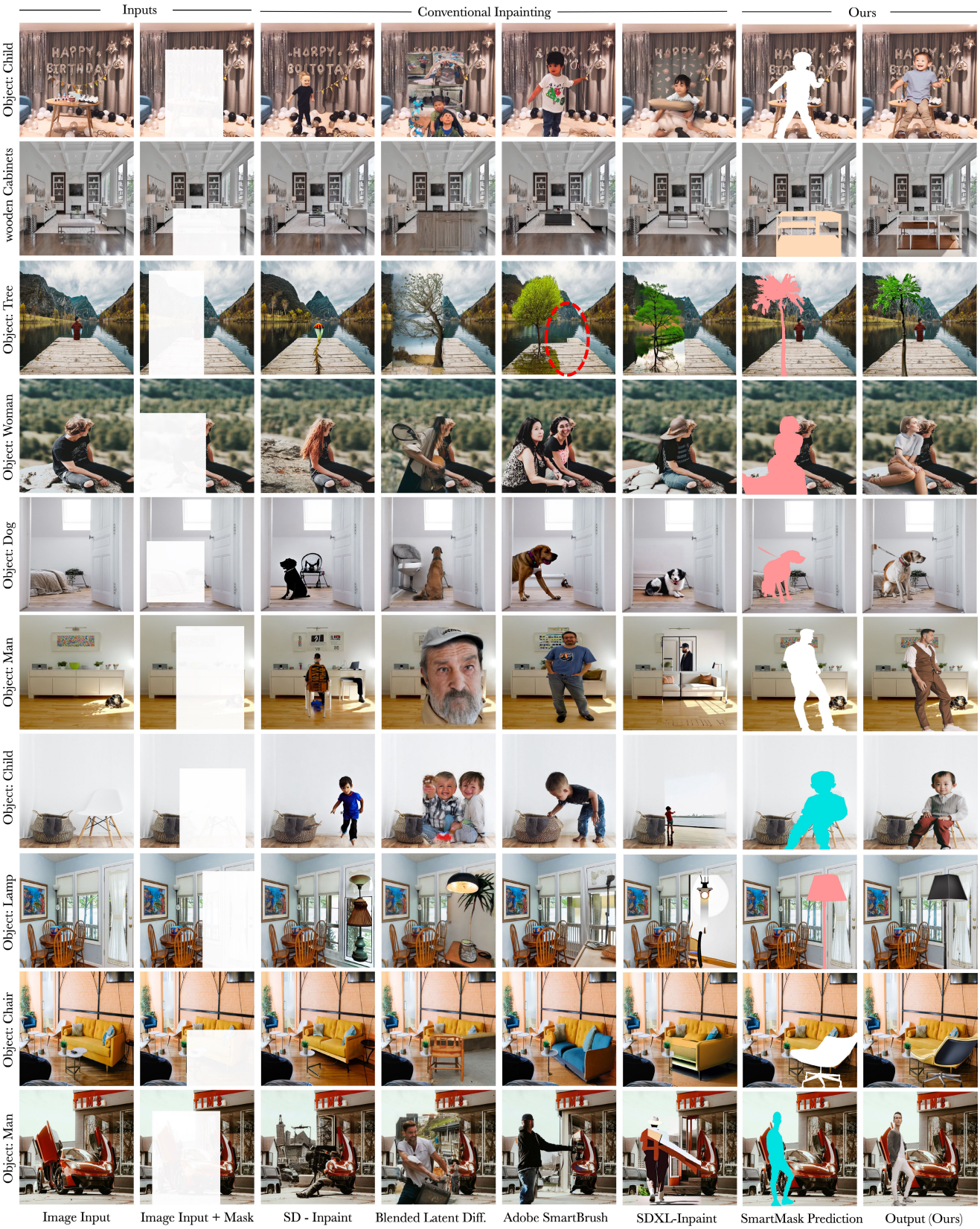

Figure 3. ***Additional Results for Single Object-Insertion.*** We observe that as compared to with state-of-the-art image inpainting [3, 16, 19, 23] methods, *SmartMask* allows the user to perform object insertion while better preserving the background around the inserted object.

tion, where it automatically provides diverse suggestions for inserting the target-object in the input image.

Results are shown in Fig. 4, Fig. 5. We observe that unlike prior image-inpainting methods which rely on user-provided masks for determining object location and scale, *SmartMask* also allows for *mask-free object insertion*. This allows the user to generate diverse object insertion suggestions for putting the target object (*e.g.*, *ship* in row-1 Fig. 4, *chair* in row-5 Fig. 5) in the input image at different positions and scales. We also observe that the masks are generated in a scene-aware manner, and can therefore account for the existing scene elements when adding the new object. For instance, when adding *man on bed* in row-5 Fig. 4, we observe that model autoamtically predicts diverse suggestions where the man is sitting or lying down on the bed in diverse poses. Similarly, when adding *a car riding down the mountain road* in row-3 Fig. 4, the car is positioned at correct pose and angle on the road region, despite the turning and slanted nature of the road. Similarly, when adding *a lamp to a living room* in Fig. 5, we observe that *SmartMask* is able to understand the scene-geometry and is able to provide diverse suggestions for positioning the lamp both on the floor and the background cabinets.

Finally, we also observe that the object insertion suggestions are generated at different scales depending on the position within the input image. Thus, objects close to camera are added through larger masks while objects away from camera are generated using smaller masks (*e.g.*, *adding a motorbike parked beside a car* in row-4 Fig. 5).

A.4. Additional Results on Output Mask Quality

We also report additional results comparing output mask quality with Smartbrush [23] mask head which also predicts an object mask for the inserted object in addition to the inpainted output. Furthermore, we also compare the performance of *SmartMask* generated masks with Inpaint + HQSAM methods when user bounding-box guidance is present. To this end, given an input image, object description and bounding-box mask, we first use an inpainting method *e.g.* Adobe SmartBrush [23] in order to inpaint the target object in the scene. The inpainted image output and corresponding bounding box are then used as input to the recently proposed HQ-SAM [11] model in order to obtain segmentation masks for the target object.

Results are shown in Fig. 6, 7. We observe that SmartBrush mask head [23] only generates very coarse masks for the target object during inpainting. Furthermore, the SmartBrush masks often have artifacts *e.g.*, *woman* in Fig. 6 and *living room chair in* Fig. 7. Passing the output of the SmartBrush inpainting to HQ-SAM [11] instead generates more finer quality masks. However, the generated masks are still fairly coarse and still have noticeable artifacts. For instance, when adding a tree to an outdoor scene with a lake (Fig. 6), we observe that SmartBrush + HQSAM generates very coarse masks for the tree region. In contrast, *SmartMask* is able to generate significantly more detailed and diverse masks for the target object (*tree*). Similarly, even when adding a *living room chair* to a living room scene (refer Fig. 7), we observe that SmartBrush + HQSAM masks typically add very coarse masks for the target chair/couch. In contrast, *SmartMask* is able to generate diverse variations with more finegrain details *e.g.*, chair structure, pose, style *etc*. for inserting the target object in the original image.

In addition to poor quality mask outputs (refer Fig. 6, 7), we observe that Inpaint + HQSAM methods (*i.e*. inpainting first and then using HQSAM to obtain target object masks) can also lead to scene-unaware masks (refer Fig. 8). For instance, in Fig. 8 we observe that when inpainting the target object (*i.e*. *woman*) in the bounding box area, Adobe SmartBrush [23] modifies the background to also add a chair (row-2). The use of HQSAM [11] on the inpainted outputs thus leads to masks which portray the woman as sitting in the air in the original image (row-3). In contrast, we observe that *SmartMask* is able to generate better quality scene-aware masks for inserting the target object where the *woman is either sitting on the floor* or *standing in the provided mask area*, thereby facilitating better object insertion performance.

B. Experiment Details

In this section, we provide further information regarding the implementation of our approach (refer App. B.1) as well as additional details on quantitative experiments used while reporting results in the main paper (refer App. B.2).

B.1. Implementation details

Data Preparation for Mask-Free Insertion. The key idea of *SmartMask* is to leverage semantic amodal segmentation datasets in order to obtain high-quality paired training annotations for mask-free object insertion. During training, given an image $\mathcal{I}$, with a sequence of ordered amodal semantic instance maps $\{A_1, A_2 \dots A_n\}$ and corresponding semantic object labels $\{\mathcal{O}_1, \mathcal{O}_2 \dots \mathcal{O}_n\}$, we first compute an intermediate layer semantic map as,

$$S_k = f_{layer}(\{A_1, A_2 \dots A_k\}) \quad where\ k \in [1, n]. \quad (1)$$

where k is randomly chosen from $[1, n]$ and f_{layer} is a layering operation which stacks the amodal semantic segmentation maps from $i \in [1, k]$ in an ordered manner (please refer Fig. 2 main paper) as follows,

$$f_{layer}(A_1, A_2, \dots, A_k) = \bigoplus_{i=1}^{k} A_i \odot h(\mathcal{O}_i) \quad \text{for } i \in [1, k]$$

where $\bigoplus$ represents the stacking operation and $h(\mathcal{O}_i)$ is the *rgb* encoding for the corresponding object description

Figure 4. ***SmartMask for mask-free object insertion.*** We observe that unlike prior image-inpainting methods which rely on user coarse masks for object location and scale, *SmartMask* also allows for *mask-free object insertion*. This allows the user to generate diverse object insertion suggestions for putting the target object (*e.g.*, *ship*) in the input image at different positions and scales. Note that the masks are generated in a scene-aware manner, and can therefore account for the existing scene elements (*e.g.*, *man lying on bed* in row-5, *car riding down the road* in row-3 *etc.*). Also notice that the object insertion suggestions are generated at different scales: thus objects close to camera are larger and away from camera are smaller (*e.g.*, *motorbike parked beside a car* in row-4).

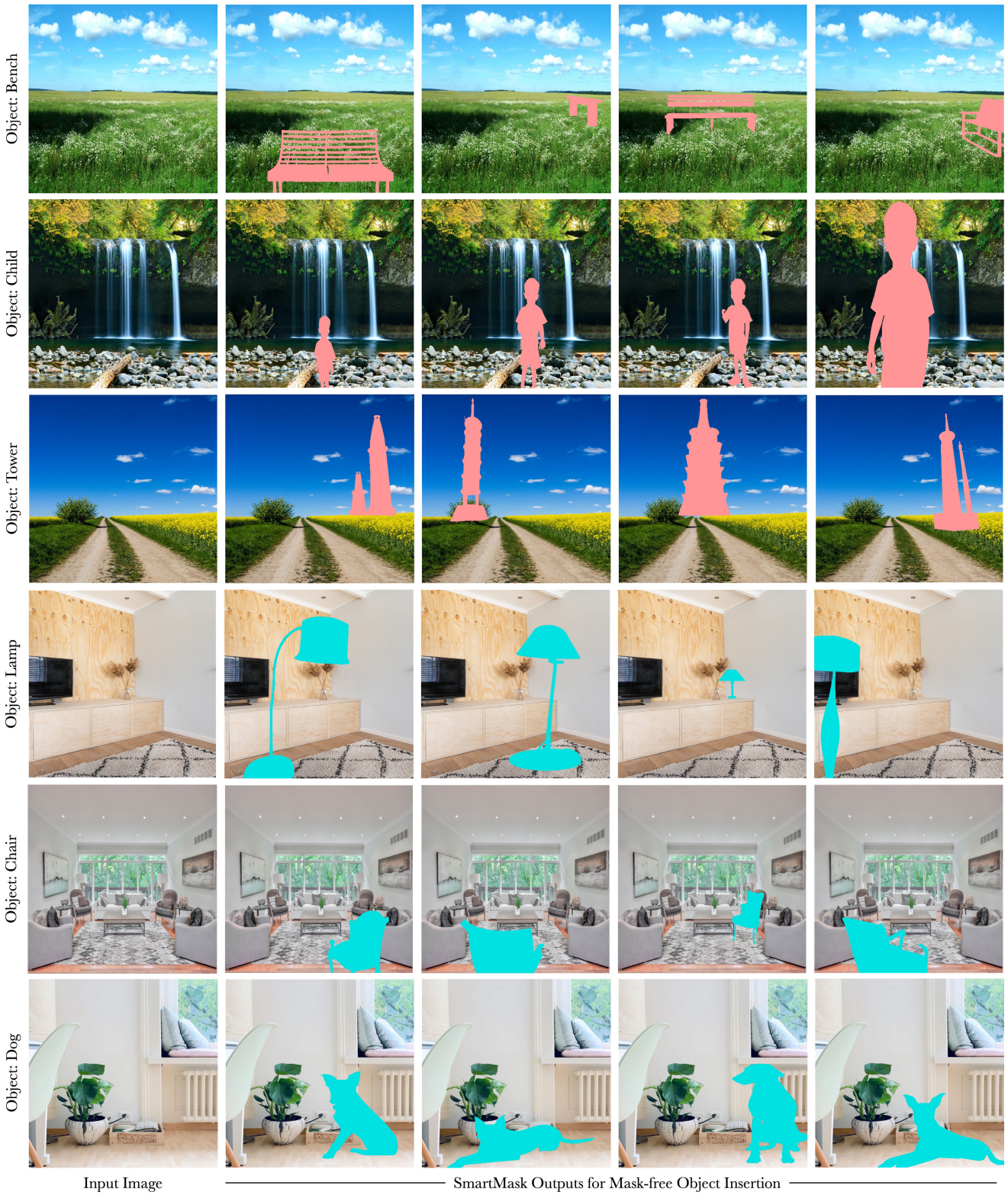

Figure 5. ***SmartMask for mask-free object insertion.*** We observe that unlike prior image-inpainting methods which rely on user coarse masks for object location and scale, *SmartMask* also allows for *mask-free object insertion*. This allows the user to generate diverse object insertion suggestions for putting the target object (*e.g.*, *bench* in row-1, *dog* in row-6) in the input image at different positions and scales. Note that the masks are generated in a scene-aware manner, and can therefore account for the existing scene elements (*e.g.*, *lamp in living room* in row-4, *chair* at different empty positions in row-5 *etc.*). Also notice that the object insertion suggestions are generated at different scales: thus objects close to camera are larger and away from camera are smaller (*e.g.*, *child near a waterfall* in row-2).

Figure 6. ***Analysing output mask quality.*** We observe that SmartBrush mask-head predictions [23] and Inpaint+HQSAM outputs (*i.e.*, inpainting first and then using HQSAM [11] typically lead to coarse masks for the target object. Furthermore, the generated masks can often have artifacts in the target regions. In contrast, *SmartMask* is able to generate diverse mask suggestions with more finegrain details (e.g., *women details* in example-1, *tree structure* in example-2) for inserting the target object in the original image.

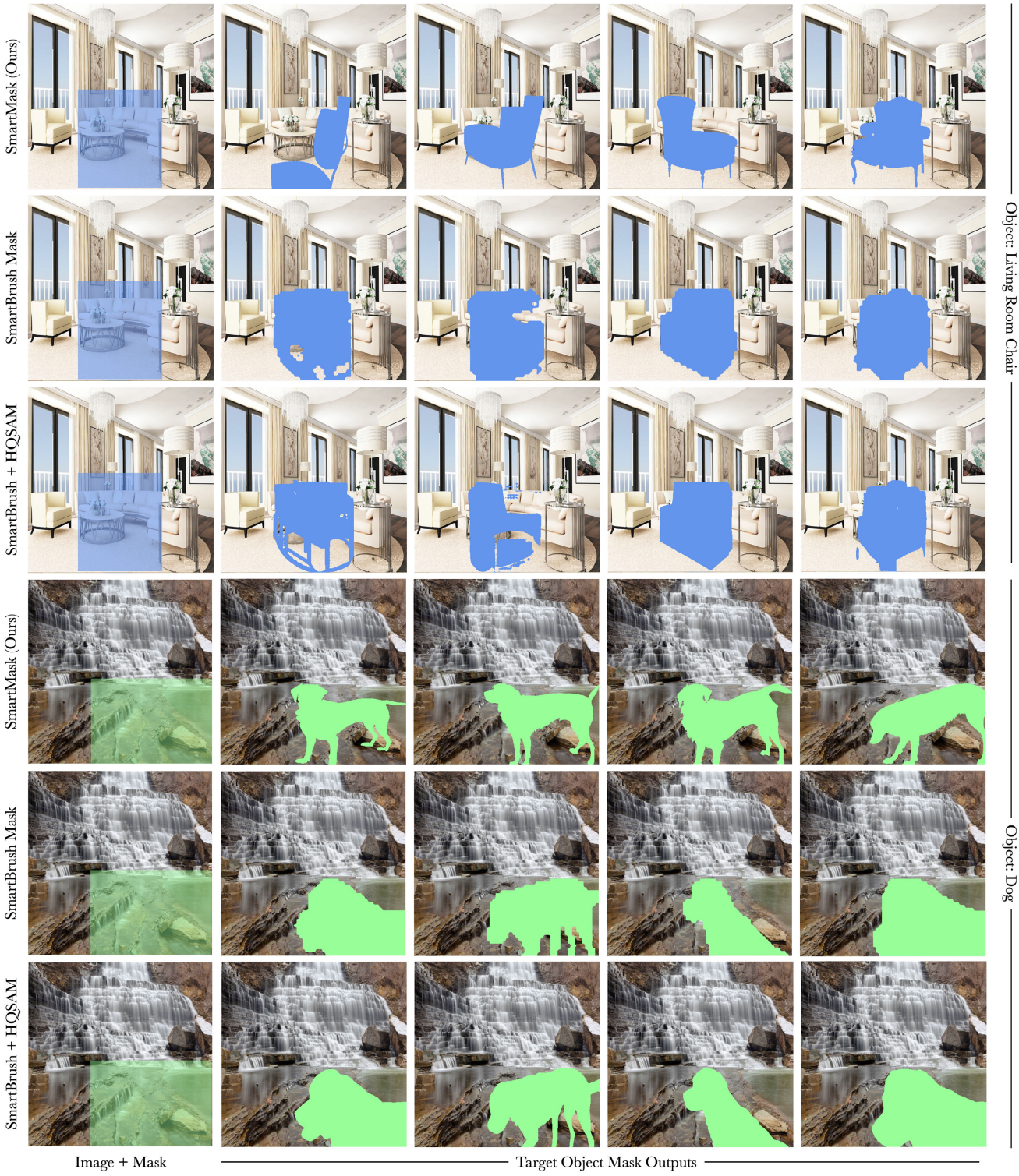

Figure 7. ***Analysing output mask quality.*** We observe that SmartBrush mask-head predictions [23] and Inpaint+HQSAM outputs (*i.e.*, inpainting first and then using HQSAM [11] typically lead to coarse masks for the target object. Furthermore, the generated masks can often have artifacts in the target regions. In contrast, *SmartMask* is able to generate diverse mask suggestions with more finegrain details (e.g., *living room chair* in example-1, *dog details* in example-2) for inserting the target object in the original image.

Figure 8. ***Limitation of Inpaint + HQSAM.*** In addition to poor quality mask outputs (refer Fig. 6, 7), we observe that Inpaint+HQSAM methods (*i.e.* inpainting first and then using HQSAM to obtain target object masks) can also lead to scene-unaware masks. For instance, we observe that when inpainting the target object (*i.e. woman*) in the bounding box area, Adobe SmartBrush [23] modifies the background to also add a chair (row-2). The use of HQSAM [11] on the inpainted outputs thus leads to masks which portray the woman as sitting in the air in the original image (row-3). In contrast, we observe that *SmartMask* is able to generate better quality scene-aware masks for inserting the target object where the *woman is either sitting on the floor* or *standing in the provided mask area*.

$\mathcal{O}_i$. The *SmartMask* model $\mathcal{D}_\theta$ is then trained to as input the above computed intermediate semantic layer map S_k, semantic label description for next object $\mathcal{T}_{obj} \leftarrow \mathcal{O}_{k+1}$, overall caption $\mathcal{T}_{context} \leftarrow C_\mathcal{I}$ (for image $\mathcal{I}$ computed using InstructBLIP [8]) and learns to predict the mask $\mathcal{M}_{obj}$ for the next object $\mathcal{O}_{k+1}$ as,

$$\mathcal{M}^{pred}_{obj} = \mathcal{D}_\theta(\mathcal{E}_\phi(\mathcal{S}_I), \mathcal{T}_{obj}, \mathcal{T}_{context}), \quad (2)$$

$$s.t. \quad \mathcal{M}^{gt}_{obj} = S_k \odot (1 - A_{k+1}) + A_{k+1} \quad (3)$$

where the ground-truth model output $\mathcal{M}^{gt}_{obj}$ is modelled as the target object mask A_{k+1} stacked on top of input map S_k which helps provide the model better spatial context for target mask generation during the reverse diffusion process.

Data Adaptation for Mask Control. In addition to mask-free object insertion, *SmartMask* also allows the user to provide an additional guidance input $\mathcal{G}_{obj}$ to control the better details of the output mask. During training, this is achieved through a simple data-adaptation strategy which

replaces input S_k to *SmartMask* as,

$$\tilde{S_k} = g(S_k, G_{obj}) = S_k \odot (1 - \alpha\, G_{obj}) + \alpha\, G_{obj}, \quad (4)$$

where G_{obj} is the additional guidance input (*e.g.*, bounding box mask, coarse scribbles *etc.*) provided by the user and $\alpha = 0.7$ helps add additional guidance while still preserving the content of the original input S_k after data adaptation.

In this paper, we mainly consider four main guidance inputs G_{obj} for additional mask control while adapting SmartMask. *1) Mask-free guidance:* $G_{obj} = \mathbf{0}^{H,W}$ which prompts the model to suggest fine-grained masks for object insertion at diverse positions and scales. *2) Bounding-box guidance:* we set G_{obj} as a binary mask corresponding to bounding-box for target object mask A_{k+1}. *3) Coarse Spatial Guidance:* Given bounding coordinates $\mathbf{b}_{k+1} = \{x_{min}, y_{min}, x_{max}, y_{max}\}$ for the target object mask A_{k+1}, we first compute a randomly perturbed bbox location as,

$$\tilde{\mathbf{b}}_{k+1} \leftarrow \mathbf{b}_{k+1} + \mathbf{p}_{k+1}, where \quad \mathbf{p}_{k+1} \in \mathbf{R}^4 \quad (5)$$

where $\mathbf{p}_{k+1} \sim U[-20, 20]$ is sampled from a uniform distribution, and randomly perturbs the original bounding box $\mathbf{b}_{k+1}$. The guidance input G_{obj} is then computed by applying a Gaussian-blur operation to a binary bounding-box mask for the perturbed bounding-box coordinates $\tilde{\mathbf{b}}_{k+1}$

4) User scribbles: Finally, we also allow the user to describe target object using free-form coarse scribbles, by setting G_{obj} as the output of multi-scale morphological dilation operation applied on target object mask A_{k+1}.

SmartMask Training. In order to leverage the rich generalizable prior of T2I diffusion models, we use the weights from publicly available Stable-Diffusion-v1.5 model [19] in order to initialize the weights of the *SmartMask* U-Net model. Similar to [6], we modify the architecture of the U-Net model to also condition the output mask predictions on segmentation layout $\mathcal{S}_I$. The semantic object label $\mathcal{T}_{obj}$ and final-scene context $\mathcal{T}_{context}$ are jointly fed to the diffusion model by modifying the input text tokens as follows,

$$\tau_{text} = \text{CLIP}(\mathcal{T}_{obj}) + \texttt{<sep>} + \text{CLIP}(\mathcal{T}_{context}), \quad (6)$$

where <sep> is the separation token and CLIP($\mathcal{T}_{obj}$), CLIP($\mathcal{T}_{context}$) represents the CLIP tokens for the object-description $\mathcal{T}_{obj}$ and scene-context $\mathcal{T}_{context}$ respectively. The overall *SmartMask* model is then trained for a total of 100k iterations with a batch size of 192 and learning rate $1e-5$ using 8 Nvidia-A100 GPUs.

SmartMask Inference. During inference, a panoptic semantic segmentation model finetuned on the dataset in Sec. 4 of main paper is used for converting real image $\mathcal{I}$ to its semantic layout $\mathcal{S}_I$. The semantic layout $\mathcal{S}_I$ along with object description and scene context are used as input to *SmartMask* model followed by a thresholding operation to obtain the target object mask. Finally, a ControlNet-Inpaint [22, 24] model trained with SDXL backbone is used to perform precise object insertion with SmartMask outputs.

Global Planning Model. The global planning model (Sec. 3.3 main paper) directly follows the visual-instruction tuning architecture from [13]. In particular, we use the recently proposed instruction-following *vicuna-1.5* [25] model as the large-language encoder. The feature alignment process was performed for 5k training steps with a learning rate of $1e-5$. The visual projector model learned from the feature alignment step is then used for instruction-tuning for predicting potential locations for the target object. The instruction-finetuning process is performed for 60k training steps with a learning rate of $1e-6$.

B.2. Quantitative Experiments

Object Insertion. In addition to qualitative results, we also report quantitative results comparing our approach to prior works [19] for object insertion. To this end, we first collect a total of 8490 image-mask pairs where human users are shown a real image input and asked to provide a feasible bounding box mask for inserting a target semantic object (*e.g.*, man, woman, dog, tree, chair *etc.*) in the original image. The image-mask pairs along with the corresponding object description are then used to obtain object inpainting results for different baselines [4, 16, 19]. The final scene context description $\mathcal{T}_{context}$ for *SmartMask* is obtained by appending the object description in front of the scene-caption (obtained using InstructBLIP [8]) for the original image. The results for Adobe Smartbrush [23] including both in-painted outputs as well as mask-head predictions (refer Sec. A.4) are obtained directly from paper authors. The results for Adobe GenerativeFill [1] are obtained using the commercially available GenFill tool from Adobe Firefly [1]. However, since no API for the same is available, quantiative results for GenFill (Table 1 main paper) are reported using a limited subset of 200 examples.

Output Mask Quality. We also report quantitative results on output mask quality by comparing *SmartMask* generated masks with SmartBrush [23] mask-head predictions and different Inpaint+HQSAM methods (*i.e.* inpainting first and then using HQSAM to obtain target object masks). To this end, we perform a human user study where given the input-image and object description, human subjects are shown a pair of object-insertion mask suggestions (ours vs baselines discussed above). For each pair, the human participant is then asked to select the object-mask suggestion with higher quality in terms of mask details, alignment with object description and mask realism/artifacts. The user-study data was performed among 50 human participants, who were given an unlimited time in order to ensure high quality of the final results. Additionally, in order to remove data noise, we use a repeated comparison (control seed) for each user. Responses of users who answer differently to this

Figure 9. *Shadow depiction.* We observe that shadow generation remains a challenging problem for even with most state-of-the-art image inpainting methods (*e.g.*, Adobe GenFill [1] *(left)* and Ours *(right)*). Nevertheless, we find that the *SmartMask* generated precise mask can be used as input to a second shadow-generation ControlNet model [15] for better quality shadow generation for the inserted object.

repeated seed are discarded while reporting the final results.

C. Additional Related Work

Bounding-box based layout generation methods. In addition to the semantic-layout to image generation works [5, 7, 12, 21, 24] discussed in Sec. 2 of the main paper, the layout generation ability of the proposed approach can also be contrasted with bounding-box based specialized layout creation methods [2, 9, 10, 18, 26]. However, the generated layouts are represented by coarse bounding-box locations. In contrast, iterative use of the proposed *SmartMask* model allows the user to control the scene layouts on a more fine-grain level including object shape / structure, occlusion relationships, location *etc*. Furthermore, as illustrated in Fig. 7b (main paper), we note that *SmartMask* generated layouts are highly controllable and allow for a range of custom operations such as adding, removing, modifying or moving objects through simple layer manipulations.

Object stamp and shape generation has also been studied in the context of object insertion and semantic layout generation. For instance, [14] propose a GAN-based approach for generating object stamps before generating their texture for object insertion. However, the same requires costly data collection and training a separate object-stamp and texture generation model for each semantic class, which can be quite time-consuming for practical applications.

D. Discussions and Limitations

While the proposed approach allows for better quality object insertion and layout control, it still has some limitations. *First,* recall that current *SmartMask* model is trained to predict object insertion suggestions based on the semantic layout $\mathcal{S}_I$ of the input image $\mathcal{I}$. While this allows us to leverage large-scale semantic amodal segmentation datasets [17, 27] for obtaining high quality paired annotations during training, the use of semantic layout input for target mask prediction can be also be limiting as the semantic layout $\mathcal{S}_I$ typically has less depth context as opposed to the original image $\mathcal{I}$. In future, using a ControlNet generated S2I image as pseudo-label can help better train the model to directly predict the target object masks from the original image $\mathcal{I}$.

Second, we note that in order to facilitate background preservation and mask-free object insertion, the current *SmartMask* model is trained on a semantic amodal segmentation dataset consisting a total of 32785 diverse real world images (with $\sim$ 0.75M different object instances). In contrast, typical object inpainting models such as Adobe SmartBrush [23], SDInpaint [19, 22] are trained on datasets [20] which are orders of magnitude larger in comparison (*e.g.* Adobe SmartBrush [23] and SDInpaint [19] are trained on 600M samples from the LAION-Aesthetics-v2 5+ dataset [20]). While utilizing the generalizable prior of a pretrained Stable-Diffusion v1.5 model [22] allows our approach to generalize across diverse semantic object categories (*e.g.* mountains, building/towers, humans, dogs, cats, clouds, trees, furniture, appliances *etc*.) with limited data, generating precise object masks for out-of-distribution semantic object labels *e.g.*, dragons, tigers *etc*. remains challenging. In future, the use of larger training datasets and stronger prior model (SDXL [16]) can help alleviate this problem.

Finally, we note that similar to prior inpainting methods [1, 16, 19, 23], accurate shadow-generation around the inserted object remains a challenging problem. For instance, in Fig. 9, when adding a man in front of a house on a sunny day, we observe that both Adobe GenFill [1] and *SmartMask* lead to limited shadow depiction around the inserted object. Nevertheless, we find that the *SmartMask* generated precise mask can be used as input to a second shadow-generation ControlNet model [15] for better quality shadow generation for the inserted object. That said, we note that precise shadow generation for the inserted object remains a challenging problem (with both prior work and ours). However, the same is out of scope of this paper, and we leave it as a direction for future research.

References

[1] Adobe. Adobe firefly – generative ai for everyone, 2023. 1, 2, 3, 11, 12

[2] Diego Martin Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13642–13652, 2021. 12

[3] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics (TOG)*, 42(4):1–11, 2023. 4

[4] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 11

[5] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 12

[6] 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. 11

[7] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. *arXiv preprint arXiv:2304.03373*, 2023. 12

[8] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 10, 11

[9] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *arXiv preprint arXiv:2305.15393*, 2023. 12

[10] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014, 2021. 12

[11] 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. 5, 8, 9, 10

[12] Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *ICCV*, 2023. 12

[13] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 11

[14] Youssef Alami Mejjati, Zejiang Shen, Michael Snower, Aaron Gokaslan, Oliver Wang, James Tompkin, and Kwang In Kim. Generating object stamps. *arXiv preprint arXiv:2001.02595*, 2020. 12

[15] Li Niu, Wenyan Cong, Liu Liu, Yan Hong, Bo Zhang, Jing Liang, and Liqing Zhang. Making images real again: A comprehensive survey on deep image composition. *arXiv preprint arXiv:2106.14490*, 2021. 12

[16] 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, 4, 11, 12

[17] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2019. 12

[18] Leigang Qu, Shengqiong Wu, Hao Fei, Liqiang Nie, and Tat-Seng Chua. Layoutllm-t2i: Eliciting layout guidance from llm for text-to-image generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 643–654, 2023. 12

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1, 2, 3, 4, 11, 12

[20] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 12

[21] Jaskirat Singh, Stephen Gould, and Liang Zheng. High-fidelity guided image synthesis with latent diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5997–6006. IEEE, 2023. 12

[22] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 1, 2, 3, 11, 12

[23] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22428–22437, 2023. 1, 4, 5, 8, 9, 10, 11, 12

[24] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1, 2, 3, 11, 12

[25] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 11

[26] Xinru Zheng, Xiaotian Qiao, Ying Cao, and Rynson WH Lau. Content-aware generative modeling of graphic design layouts. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 12

[27] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1464–1472, 2017. 12