SpaText: Spatio-Textual Representation for Controllable Image Generation — Supplementary Material

Omri Avrahami[1,2] Thomas Hayes[1] Oran Gafni[1] Sonal Gupta[1]
Yaniv Taigman[1] Devi Parikh[1] Dani Lischinski[2] Ohad Fried[3] Xi Yin[1]

[1]Meta AI [2]The Hebrew University of Jerusalem [3]Reichman University

Acknowledgments We thank Uriel Singer, Adam Polyak, Yuval Kirstain, Shelly Sheynin and Oron Ashual for their valuable help and feedback. This work was supported in part by the Israel Science Foundation (grants No. 1574/21, 2492/20, and 3611/21).

A. Additional Examples

In Figures 1, 2, 3, 4 and 5 we provide additional results from our model. In Figures 6 and 7 we provide additional examples for the mask insensitivity of our method. In Figures 8 and 9 we show the fine-grained control that is achievable via the multi-scale version of our method. In Figure 10 we provide additional limitations of our method.

B. Implementation Details

In the following section, we describe the implementation details that were omitted from the main paper. In Appendix B.1 we start by describing the diffusion models implementation details. Then, in Appendix B.2 we describe the implementation details of our spatio-textual representation. Later, in Appendix B.3 we describe the implementation details of the baselines and how we adapt them to our problem setting. Afterwards, in Appendix B.4 we describe the implementation details of the automatic input creation process that we used to compute our automatic metrics. Finally, in Appendix B.5 we describe the details of the user study.

B.1. Diffusion Models Implementation Details

We based our approach on two state-of-the-art diffusion-based text-to-image models: DALL·E 2 [34] and Stable Diffusion [37]. We trained these models on a custom-made dataset of 35M image-text pairs, following Make-A-Scene [9].

B.1.1 DALL·E 2 Implementation Details

Since the implementation of DALL·E 2 is not available to the public, we re-implemented it following the details included in their paper [34]. This model consists of the following submodules, given an (x, y) image-text pair:

- **A decoder model D**: that is trained to translate $\text{CLIP}_{\text{img}}(x)$ into a 64×64 resolution image x.
- **A super-resolution model SR**: that is trained to upsample the 64×64 resolution image x into 256×256.
- **A prior model P**: that is trained to translate the tuples $(\text{CLIP}_{\text{txt}}(y), \text{BytePairEncoding}(y))$ into $\text{CLIP}_{\text{img}}(x)$.

Concatenating the above three models yields a text-to-image model $SR \circ D \circ P$.

In order to adapt the model to the task of text-to-image generation with sparse scene control, we chose to fine-tune the decoder D. For the fine-tuning we used the standard simple loss variant of Ho *et al.* [13]:

$$L_{\text{simple}} = E_{t,x_0,\epsilon}\left[||\epsilon - \epsilon_\theta(x_t, \text{CLIP}_{\text{img}}(x_0), ST, t)||^2\right] \tag{1}$$

where ϵ_θ is a UNet [23] model that predicts the added noise at each time step t, x_t is the noisy image at time step t and ST is our spatio-textual representation. To this loss, we added the same variational lower bound (VLB) loss as in [26] to get the total loss of:

$$L_{hybrid} = L_{simple} + \lambda L_{VLB} \tag{2}$$

we set $\lambda = 0.001$ in our experiments. We used Adam optimizer [17] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with learning rate 6×10^{-5} for 64,000 iterations.

During inference, we utilize composition of the CLIP text encoder CLIP_{txt} and the prior model P to infer the CLIP image embedding for both the spatio-textural representation ST and for the global text prompt $P \circ \text{CLIP}_{\text{txt}}(t_{\text{global}})$. We used the DDIM [38] inference method with a different number of inference steps for each component: 50 steps for the prior model, 250 for the decoder, and 100 for the super resolution model.

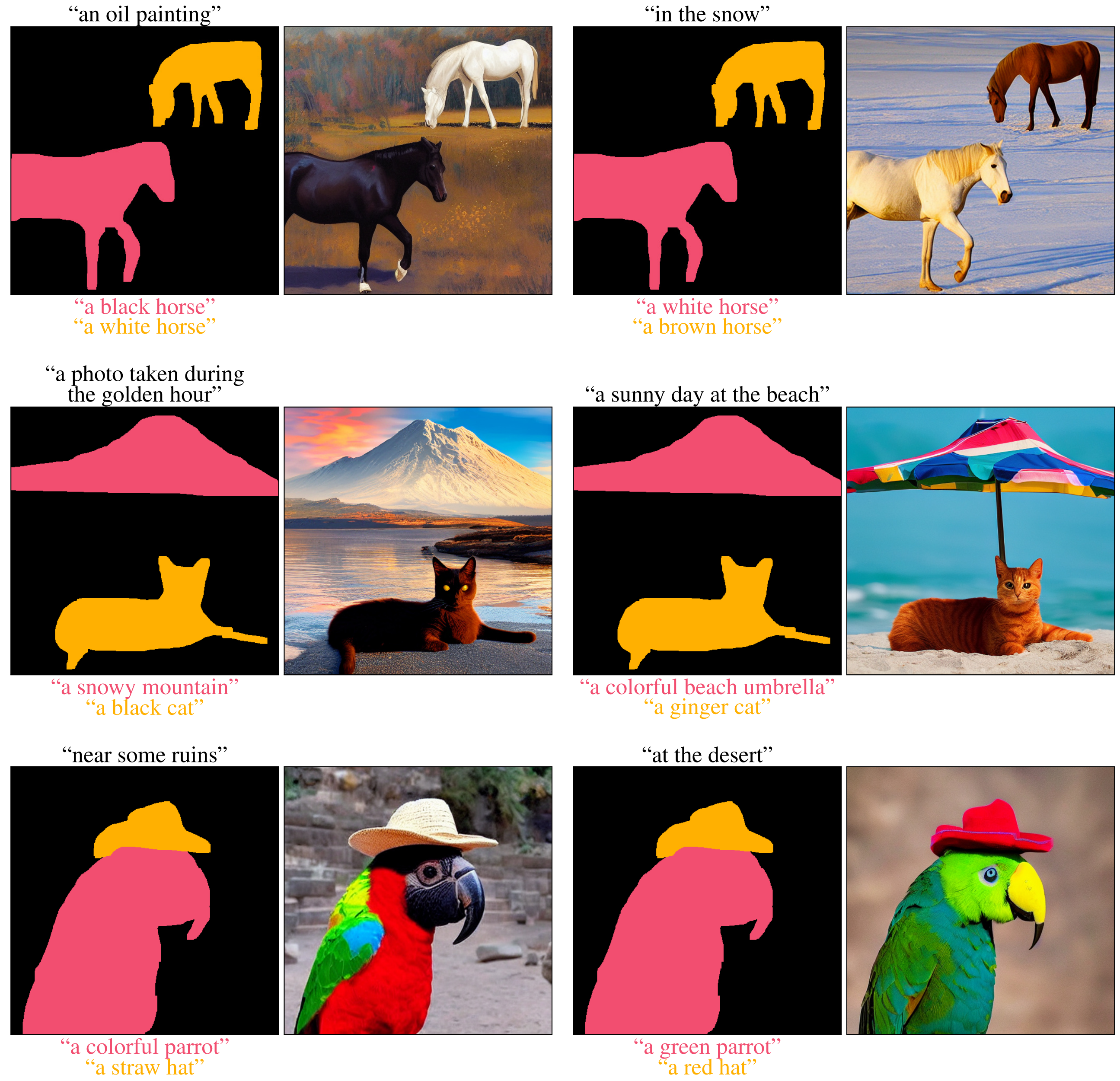

Figure 1. **Additional examples of our method:** Each pair consists of an (i) input global text (top left, black), a spatio-textual representation describing each segment using free-form text prompts (left, colored text and sketches), and (ii) the corresponding generated image (right). As can be seen, SpaText is able to generate high-quality images that correspond to both the global text and spatio-textual representation content. Please note that the colors are for illustration purposes only, and do not affect the actual inputs.

B.1.2 Stable Diffusion Implementation Details

For Stable Diffusion [37] we used the official implementation [5] and the official pre-trained v1.3 weights from Hugging Face [7].

We followed the same training procedure as the original implementation, and adapted the latent denosing model to get as an additional input the spatio-textual representation ST with the following training loss:

$$L_{\text{LDM}} = E_{t,y,z_0,\epsilon}\left[||\epsilon - \epsilon_\theta(z_t, \text{CLIP}_{\text{txt}}(y), ST, t)||^2\right] \quad (3)$$

where z_t is the noisy latent code at time step t and y is the corresponding text prompt. We fine-tuned only the denoising model while keeping the autoencoder and CLIP_{txt} frozen. We used Adam optimizer [17] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ with learning rate 1×10^{-4} for 100,000 itera-

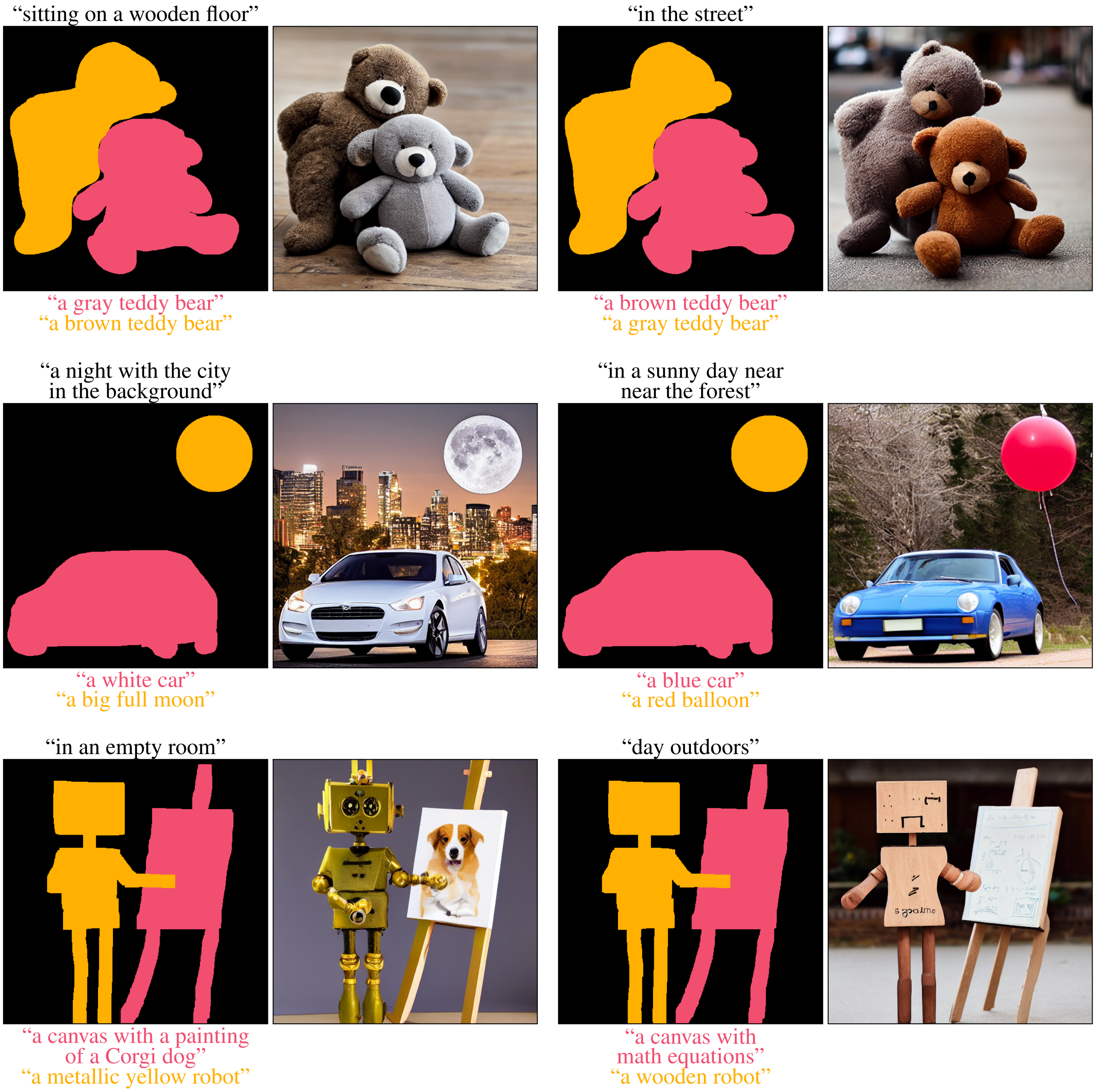
"sitting on a wooden floor"
"a gray teddy bear"
"a brown teddy bear"
"in the street"
"a brown teddy bear"
"a gray teddy bear"
"a night with the city in the background"
"a white car"
"a big full moon"
"in a sunny day near near the forest"
"a blue car"
"a red balloon"
"in an empty room"
"a canvas with a painting of a Corgi dog"
"a metallic yellow robot"
"day outdoors"
"a canvas with math equations"
"a wooden robot"

Figure 2. **Additional examples of our method:** Each pair consists of an (i) input global text (top left, black), a spatio-textual representation describing each segment using free-form text prompts (left, colored text and sketches), and (ii) the corresponding generated image (right). As can be seen, SpaText is able to generate high-quality images that correspond to both the global text and spatio-textual representation content. Please note that the colors are for illustration purposes only, and do not affect the actual inputs.

tions.

During inference, we used the DDIM [38] inference method with 50 sampling steps.

B.2. Spatio-Textual Representation Details

In order to create the spatio-textual CLIP-based representation, we used the following models:

- A pre-trained ViT-L/14 [6] variant of CLIP [33] model released by OpenAI [27].
- A pre-trained panoptic segmentation model R101-FPN from Detectron2 [44].

During the training phase, we extracted candidate segments using R101-FPN model from the Detectron2 [44] codebase model and filtered the small segments that accounted for less than 5% of the image area because their

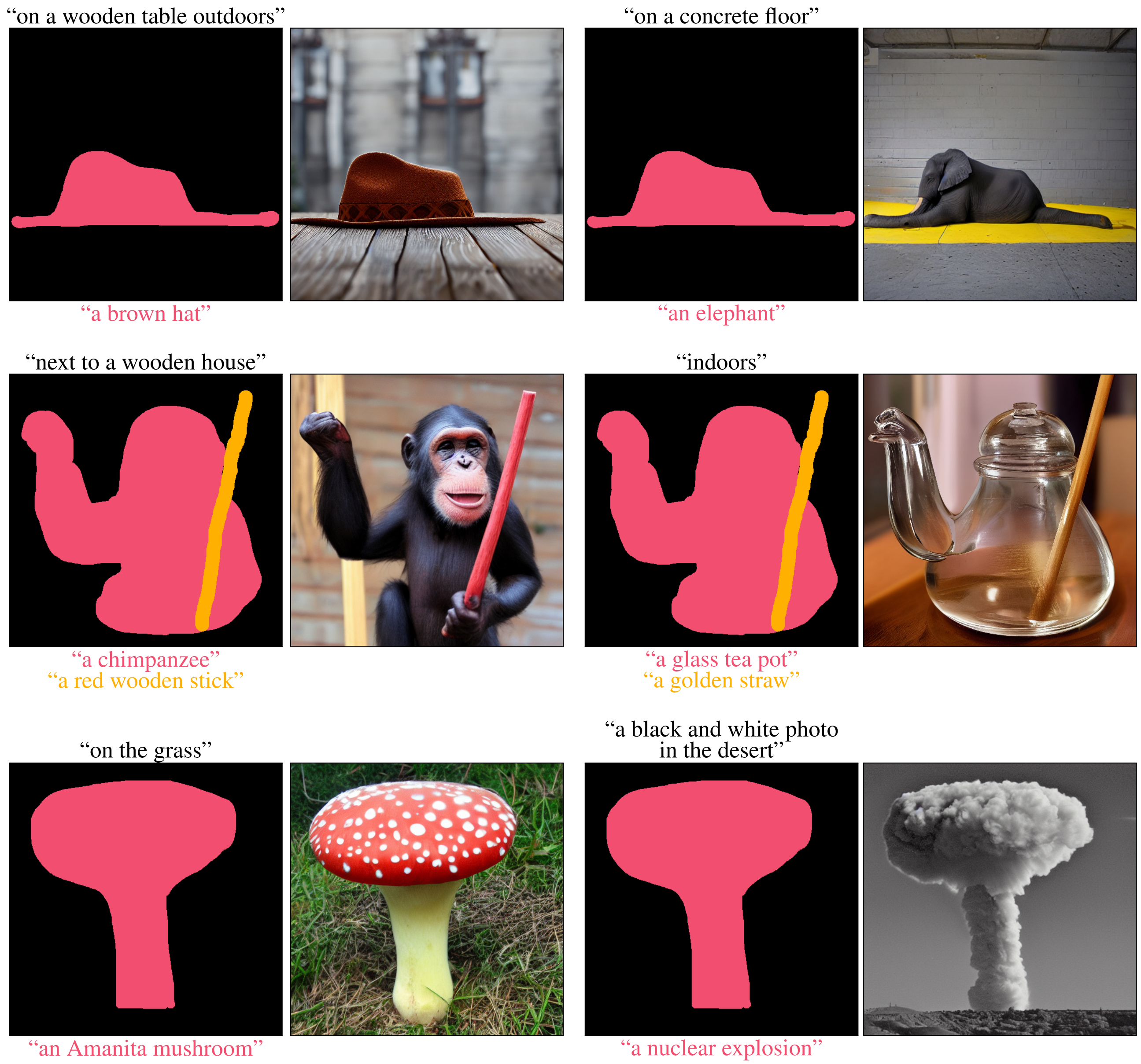

Figure 3. **Additional examples of our method:** Each pair consists of an (i) input global text (top left, black), a spatio-textual representation describing each segment using free-form text prompts (left, colored text and sketches), and (ii) the corresponding generated image (right). As can be seen, SpaText is able to generate high-quality images that correspond to both the global text and spatio-textual representation content. Please note that the colors are for illustration purposes only, and do not affect the actual inputs.

CLIP image embeddings are less meaningful for low-res images. Then, we randomly used $1 \leq K \leq 3$ segments for the formation of the spatio-textual representation.

In addition, in order to enable multi-conditional classifier-free guidance, as explained in Section 3.3 in the main paper, we dropped each of the input conditions (the global text and the spatio-textual representation) during training 10% of the time (i.e., the model was trained totally unconditionally about 1% of the time).

B.3. Baselines Implementation Details

For the No Token Left Behind (NTLB) baseline [29] we used the official PyTorch [31] implementation [1]. The original model did not support global text and was mainly demonstrated on rectangular masks. In order to adapt it to our problem setting, we added a degenerate mask of all ones for the global text. Then, we used the rest of the segmentation maps as-is, along with their corresponding text prompt. For Make-A-Scene (MAS) [9], we followed the exact implementation details from the paper.

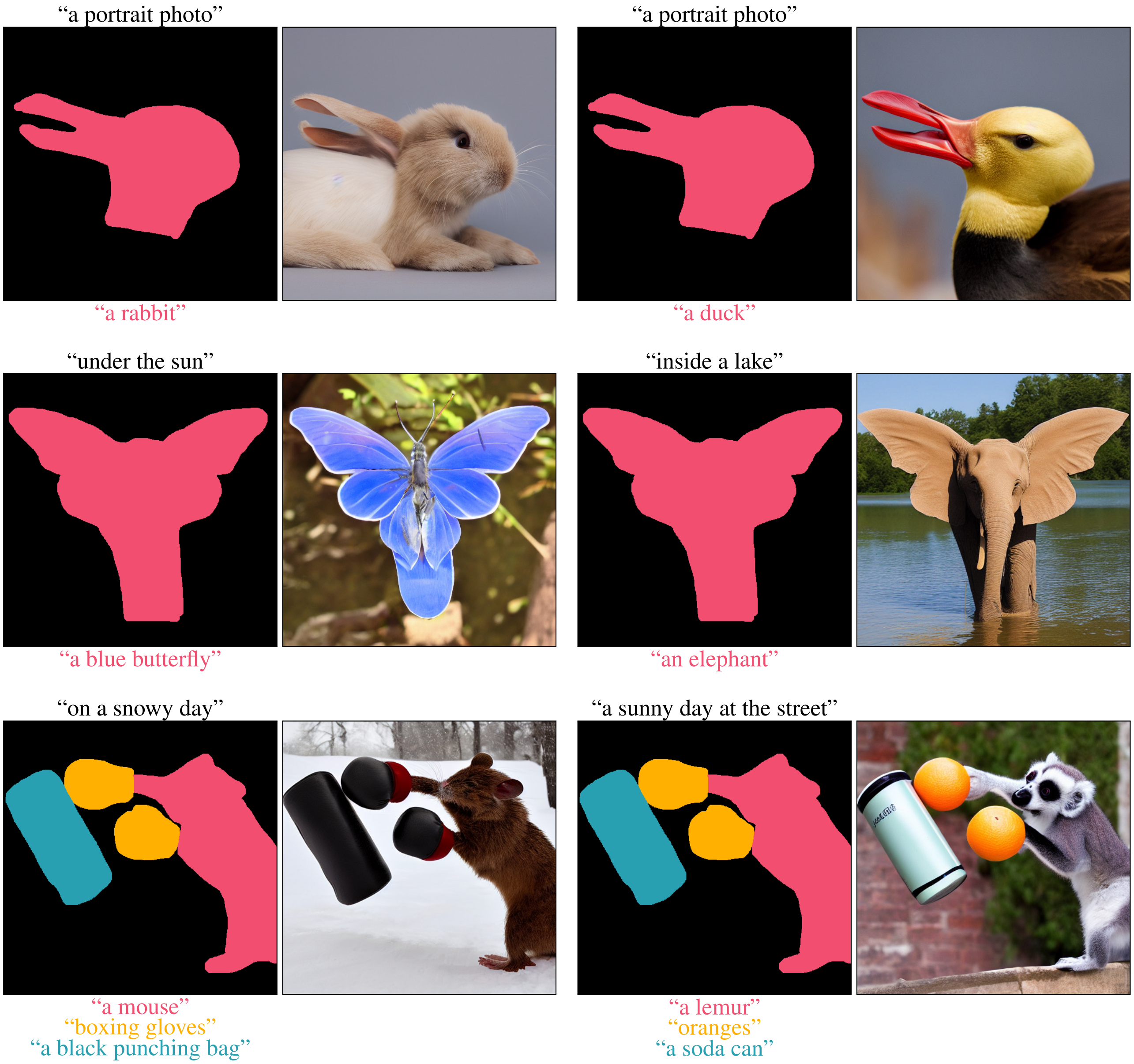

Figure 4. **Additional examples of our method:** Each pair consists of an (i) input global text (top left, black), a spatio-textual representation describing each segment using free-form text prompts (left, colored text and sketches), and (ii) the corresponding generated image (right). As can be seen, SpaText is able to generate high-quality images that correspond to both the global text and spatio-textual representation content. Please note that the colors are for illustration purposes only, and do not affect the actual inputs.

In addition, we used the official DALL·E 2 and Stable Diffusion online demos [28, 39] to generate the assets for some of the figures in this paper: Figure 2 in the main paper, and Figure 12 below.

B.4. Evaluation Dataset Details

As explained in Section 4.1 in the main paper, we proposed to evaluate our method automatically by generating a large number of coherent inputs based on natural images. To this end, we used the COCO [20] validation set that contains global text captions as well as a dense segmentation map for each image. We convert the segmentation map labels by simply providing the text “a {label}” for each segment. Then, we randomly choose a subset of size $1 \leq K \leq 3$ segments to form the sparse input. This way, we generated $30,000$ input samples for comparison. Figure 11 (top row) shows a random number of generated input samples.

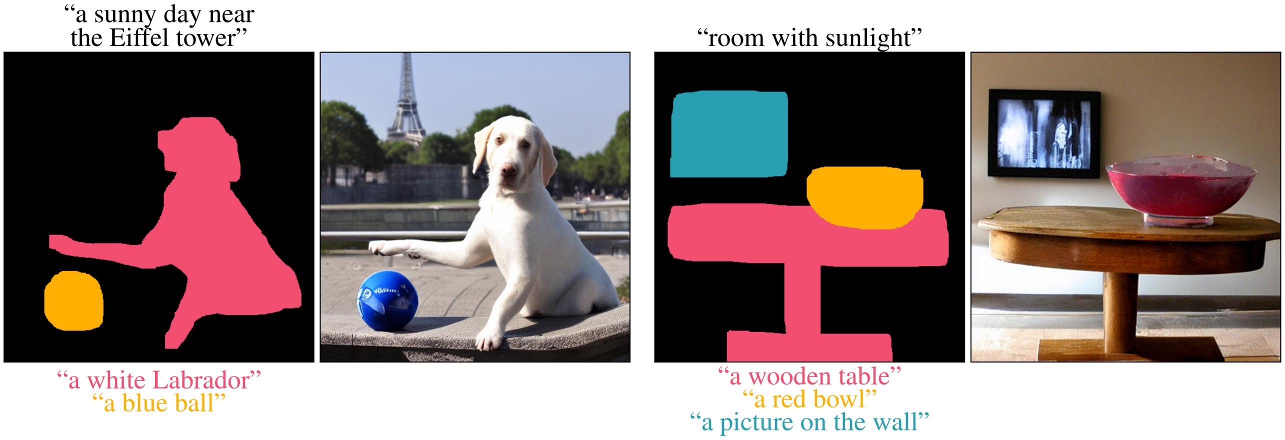

Figure 5. **Additional examples of our method:** Each pair consists of an (i) input global text (top left, black), a spatio-textual representation describing each segment using free-form text prompts (left, colored text and sketches), and (ii) the corresponding generated image (right). As can be seen, SpaText is able to generate high-quality images that correspond to both the global text and spatio-textual representation content. Please note that the colors are for illustration purposes only, and do not affect the actual inputs.

In addition, we provide in Figure 11 an additional qualitative comparison of our method against the baselines. As we can see, the latent-based variant of our method outperforms the baselines in terms of compliance with both the global and local texts, and in terms of overall image quality.

B.5. User Study

As explained in Section 4.2 in the main paper, we conducted a user study using the Amazon Mechanical Turk (AMT) platform. In each question the evaluators were asked to choose between two images in terms of (1) overall image quality, (2) text-matching to the global prompt t_{global} and (3) text-matching to the local prompts of the raw spatio-textual representation RST. For each one of those metrics, we created 512 coherent inputs automatically from COCO validation set [20] as described in Section 4.1 in the main paper and presented a pair of generated results to five raters, yielding a total of 2,560 ratings per task. For each question, the raters were asked to choose the better result of the two (according to the given criterion). We reported the majority vote percentage per question. In addition, the raters were also given the option to indicate that both models are equal, in a case which the majority vote indicated equal, or in a tie case, we divided the points equally between the evaluated models.

The questions we asked per comparison are:

- For the overall quality test — “Which image has a better visual quality?”
- For the global text correspondence test — “Which image best matches the text: {GLOBAL TEXT}”, where {GLOBAL TEXT} is t_{global}.
- For the local text correspondence test — we provided in addition one mask from the raw spatio-textual representation RST and asked “Which image best matches the text and the shape of the mask?”

B.6. Inference Time and Parameters Comparison

In Table 1 we compare the number of parameters and the inference time of the baselines and the different variants of our method. For each method, we describe its submodules and their corresponding number of parameters and inference times for a single image. As we can see, our latent-based variant is significantly faster than the rest of the baselines. In addition, it has fewer parameters than Make-A-Scene [9] and the pixel-based variant of our method.

C. Additional Experiments

In this section, we provide additional experiments that we have conducted. In Appendix C.1 we describe manual baselines that may be used to generate images with free-form textual scene control. Then, in Appendix C.2 we present a general variant for Make-A-Scene and compare it against our method. Finally, in Appendix C.3 we describe and demonstrate the local prompts concatenation trick.

C.1. Manual Baselines

In order to generate an image with free-form textual scene control, one may try to operate existing methods in various manual ways. For example, as demonstrated in Section 1 in the main paper, trying to achieve this task using an elaborated text prompt is overly optimistic. We provided additional examples in Figure 12.

Another possible option to achieve this goal it to combine a text-to-image models with a local text-driven editing method [2, 3, 34] in a multi-stage approach: at the first

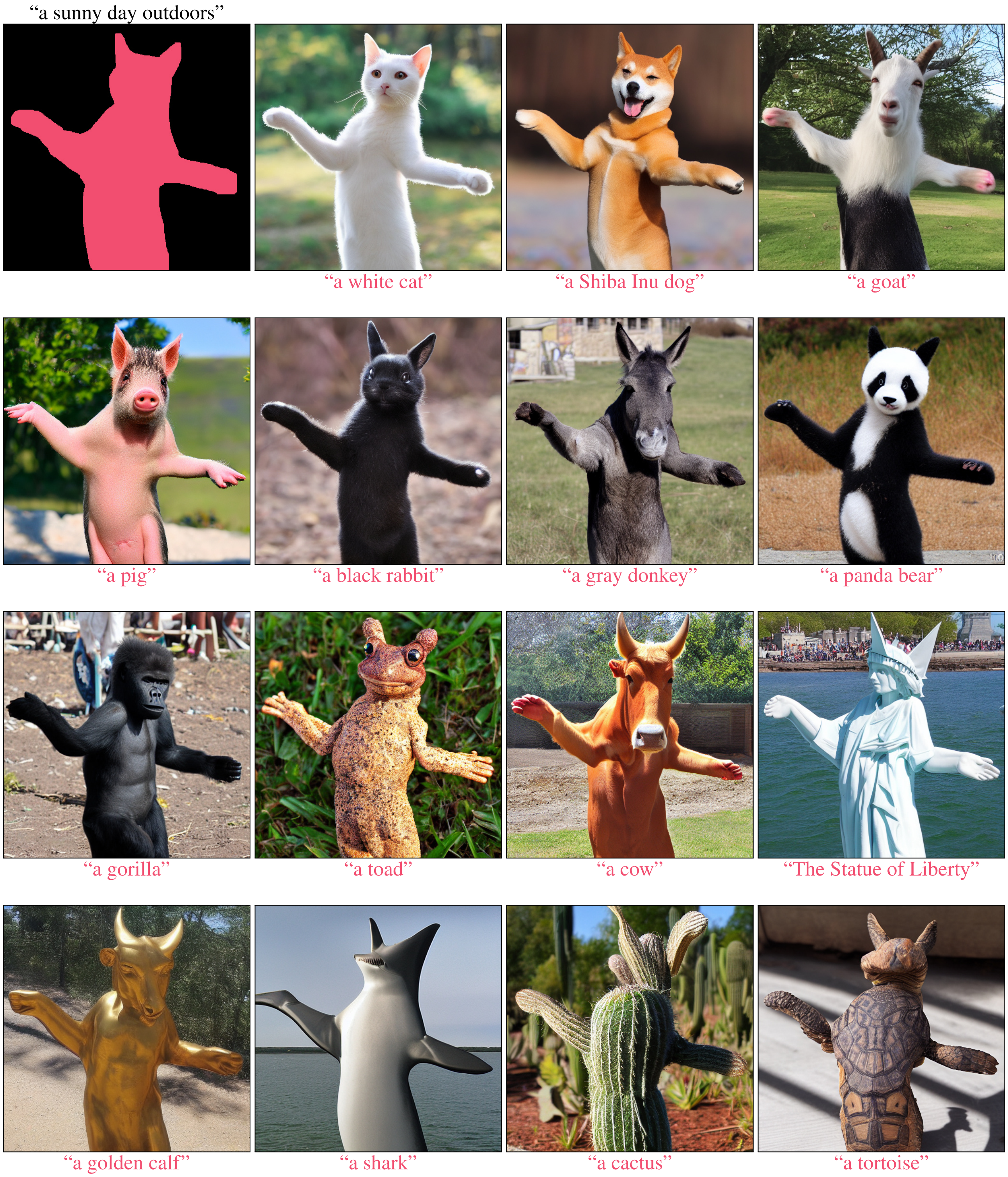

Figure 6. **Mask insensitivity:** We found that the model is relatively insensitive to inaccuracies in the input mask. Given a general animal shape mask (top left), the model is able to generate a diverse set of results driven by the different local prompts. It changes the body type according to the local prompt, while leaving the overall posture of the character intact.

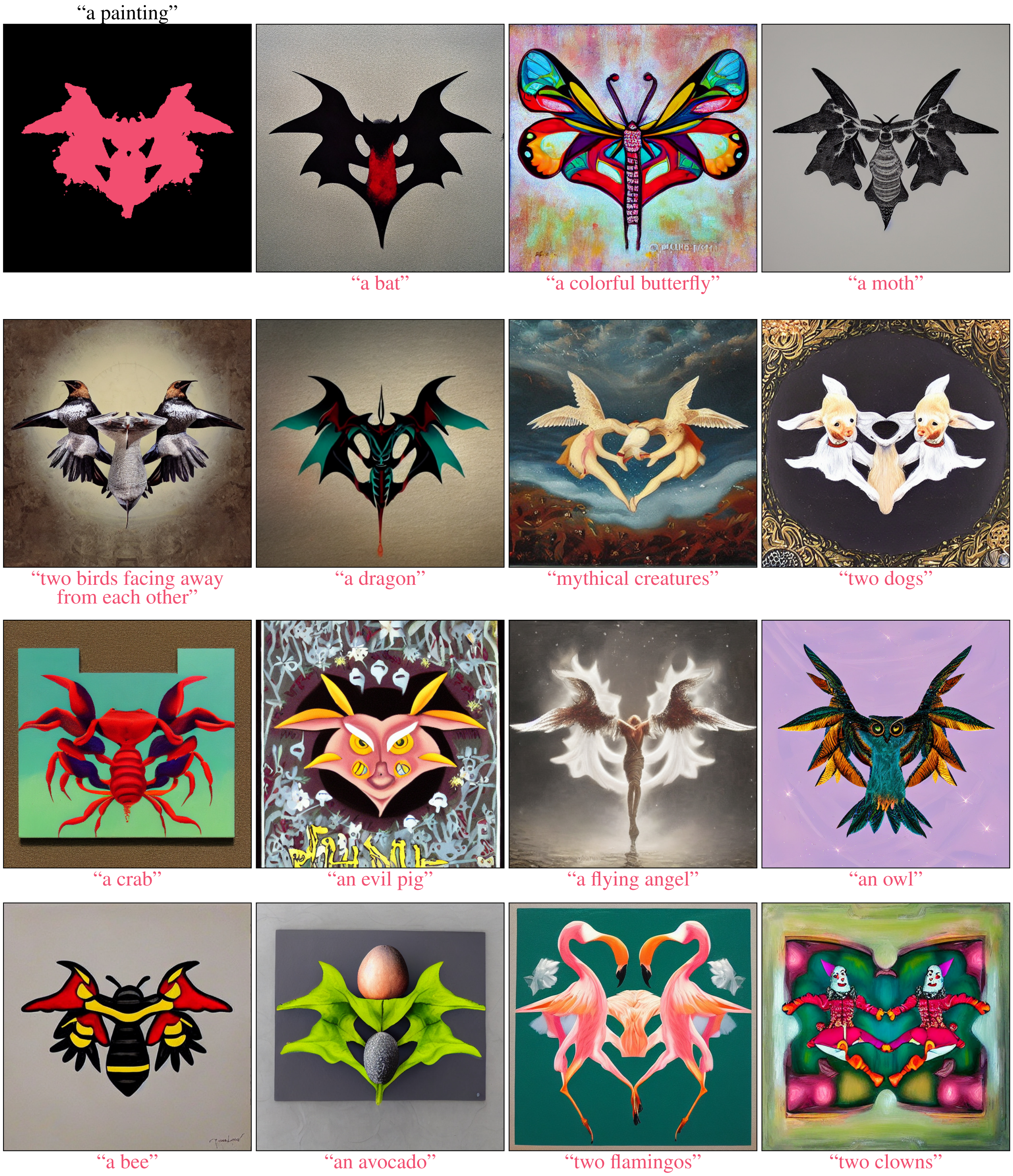

Figure 7. **Mask insensitivity:** We found that the model is relatively insensitive to inaccuracies in the input mask. Given a general Rorschach [18] mask (top left), the model is able to generate a diverse set of results driven by the different local prompts. It changes fine-details according to the local prompt, while leaving the overall general shape intact.

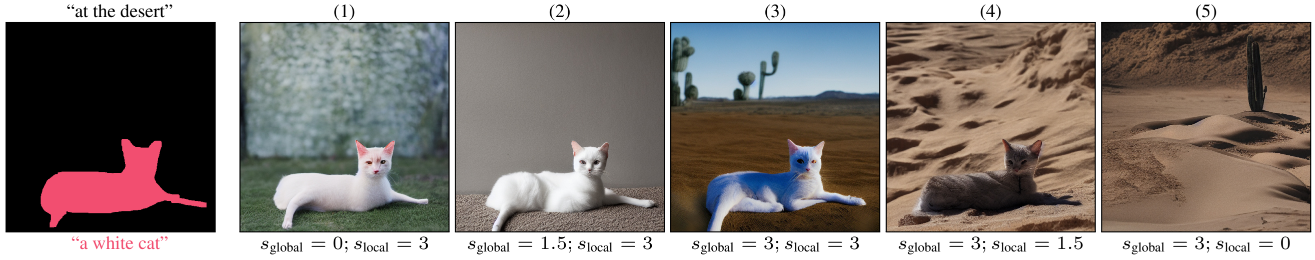
“at the desert”
“a white cat”
(1)
(2)
(3)
(4)
(5)
$s_{\text{global}} = 0; s_{\text{local}} = 3$
$s_{\text{global}} = 1.5; s_{\text{local}} = 3$
$s_{\text{global}} = 3; s_{\text{local}} = 3$
$s_{\text{global}} = 3; s_{\text{local}} = 1.5$
$s_{\text{global}} = 3; s_{\text{local}} = 0$

Figure 8. **Multi-scale control:** Using the multi-scale inference allows fine-grained control over the input conditions. Given the same inputs (left), we can use different scales for each condition. In this example, if we put all the weight on the local scene (1), the generated image contains a cat with the correct color and posture, but not at the desert. Conversely, if we place all the weight on the global text (5), we get an image of a desert with no cat in it. The in-between results correspond to a mix of conditions — in (4) we get a gray cat with slightly different posture, in (2) the cat sits on dirt, but not in the desert, and in (3) we get a white cat at the desert.

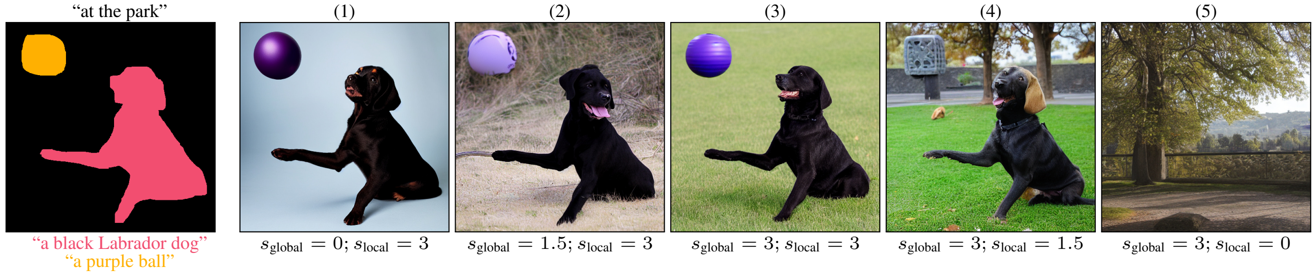
“at the park”
“a black Labrador dog”
“a purple ball”
(1)
(2)
(3)
(4)
(5)
$s_{\text{global}} = 0; s_{\text{local}} = 3$
$s_{\text{global}} = 1.5; s_{\text{local}} = 3$
$s_{\text{global}} = 3; s_{\text{local}} = 3$
$s_{\text{global}} = 3; s_{\text{local}} = 1.5$
$s_{\text{global}} = 3; s_{\text{local}} = 0$

Figure 9. **Multi-scale control:** Using the multi-scale inference allows fine-grained control over the input conditions. Given the same inputs (left), we can use different scales for each condition. In this example, if we put all the weight on the local scene (1), the generated image contains a Labrador dog and a purple ball with the correct color and posture, but not at the park. Conversely, if we place all the weight on the global text (5), we get an image of a park with no dog or ball in it. The in-between results correspond to a mix of conditions — in (4) we get a gray brick instead of a purple ball, in (2) the dog is outside but not in the park, and in (3) we get a black Labrador dog and a purple ball in the park.

stage, the user can utilize a text-to-image model to generate the background of the scene, e.g. Stable Diffusion or DALL·E 2. Then, the user can sequentially mask the desired areas and provide the local prompts using a local text-driven editing method, e.g. Blended Latent Diffusion or DALL·E 2. Figures 13 and 14 demonstrate that even though these approaches may place the object in the desired location, the composition of the entire scene looks less natural, because the model does not take into account the entire scene at the first stage, so the generated image of the background may not be easily edited for the desired composition. In addition, the objects correspond less to the local masks, especially in the DALL·E 2 case. Furthermore, the multi-stage approach is more cumbersome from the user point of view, because of its iterative nature.

Lastly, another approach is to utilize a sketch-to-image generation, as demonstrated in SDEdit [24]: the user can provide a *dense* color sketch of the scene, then noise it to a certain noise level, and denoise it iteratively using a text-to-image diffusion model. However, this user interface is different from our interface in the following aspects: (1) the user need to provide a color for each pixel, whereas in our method the user may provide a local prompt that describe other aspects that are not color-related only. In addition, (2) in this approach, the user needs to construct a *dense* segmentation map of the entire scene in advance, whereas in our method the user can provide only some of the areas and let the machine infer the rest. It is not clear how this can be done in the sketch-based approach.

C.2. Random Label Make-A-Scene Variant

In Section 4.1 in the main paper, we presented a way to adapt Make-A-Scene (MAS) [9] to our problem setting. The original Make-A-Scene work proposed a method that conditions a text-to-image model on a global text t_{global} and a *dense* segmentation map with *fixed labels*. Hence, we converted it to our problem setting of *sparse* segmentation map with *open-vocabulary local prompts* by concatenating the local texts of the raw spatio-textual representation RST into the global text prompt t_{global}.

However, the above version requires the user to provide an additional label for each segment, which is more than needed by our method and NTLB [29] baseline. Hence, we experimented with a more general version of Make-A-Scene we termed MAS (rand-label) that assigns a random label for each segment, instead of asking the user to pro-

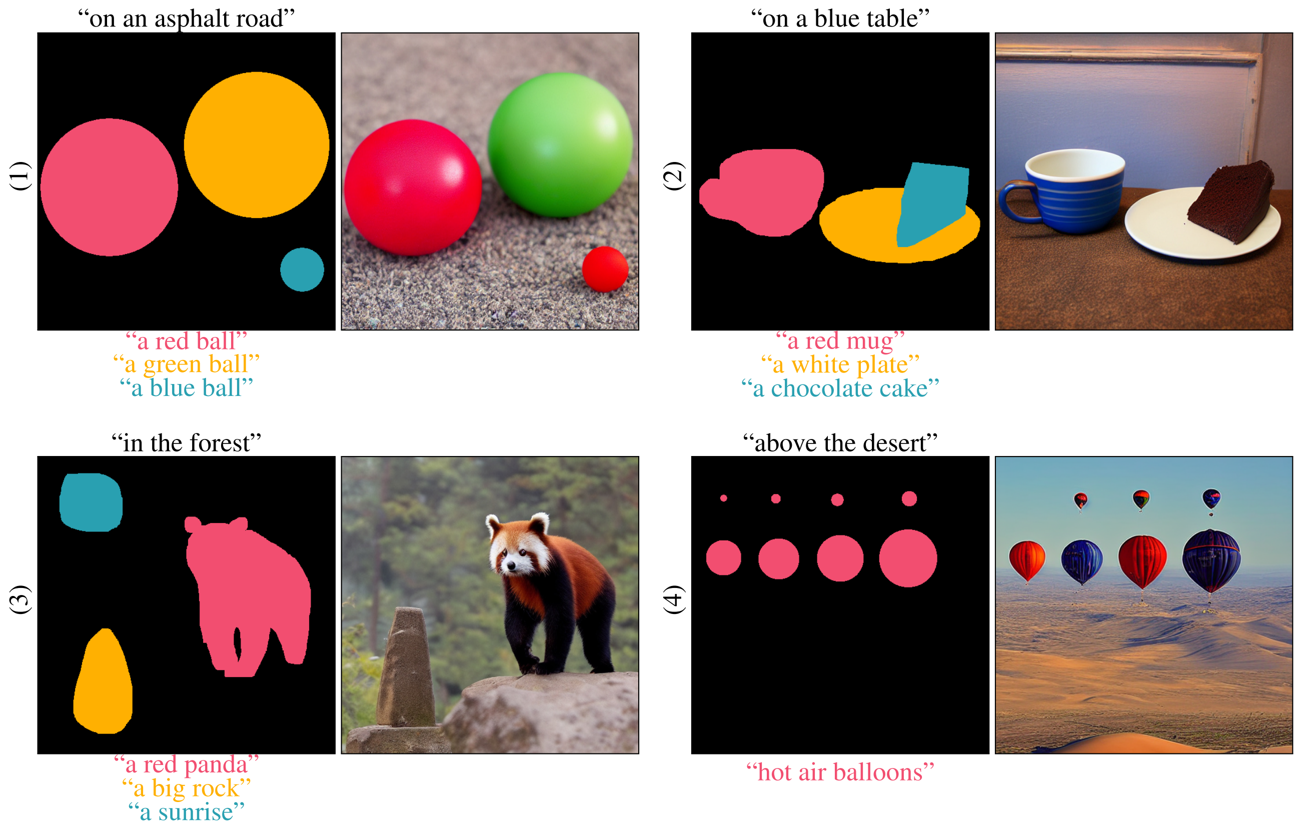

Figure 10. **Limitations:** In some cases there is a "characteristics leakage" between segments, as in example (1) where instead of a blue ball we get another red ball, or a leakage between the global text and some segments, as in example (2) where the mug is generated in a blue color originated in the global text. In other cases, the model ignores some of the objects, as the sun in example (3) and the smallest hot air balloon in example (4).

Method	**Consisting submodules**	**# Parameters (B)**	**Inference time (sec)**
No Token Left Behind [29]	CLIP (ViT-B/32) + model	0.15B + 0.08B = 0.23B	326 sec
Make-A-Scene [9]	VAE + model	0.002B + 4B = 4.002B	76 sec
SpaText (pixel) w/o prior	CLIP + model + upsample	0.43B + 3.5B + 1B = 4.93B	50 sec
SpaText (pixel) w prior	CLIP + prior + model + upsample	0.43B + 1.3B + 3.5B + 1B = 6.23B	52 sec
SpaText (latent) w/o prior	CLIP + model	0.43B + 0.87B = 1.3B	5 sec
SpaText (latent) w prior	CLIP + prior + model	0.43B + 1.3B + 0.87B = 2.6B	7 sec

Table 1. **Inference time and parameters:** we compare the number of parameters and the inference time across the baselines and the different variants (including ablations) of our method. As we can see, SpaText (latent) is significantly faster than the rest of the baselines. In addition, it has fewer parameters than Make-A-Scene [9] and the SpaText (pixel) variant of our method. The inference times reported were computed for a single image on a single V100 NVIDIA GPU.

vide an additional label. In Figure 15 we can see that this method is able to match the local prompts even with random labels. We also evaluated this method numerically using the same automatic metrics and user study protocol described in Section 4 in the main paper. As can be seen in Table 2, this method achieves inferior results compared to the version that uses the ground-truth labels in both the automatic evaluation and the user study.

C.3. Local Prompts Concatenation Trick

As described in Section 3.3 in the main paper, we noticed that the texts in the image-text pairs dataset contain elaborate descriptions of the entire scene, whereas we aim to ease the use for the end-user and remove the need to pro-

	Automatic Metrics				User Study		
Method	Global ↓	Local ↓	Local ↑	FID ↓	Visual	Global	Local
	distance	distance	IOU		quality	match	match
MAS [9]	0.7591	0.7835	**0.2984**	21.367	81.25%	70.61%	57.81%
MAS (rand-label) [9]	0.7796	0.7861	0.1544	29.593	82.81%	81.44%	76.85%
SpaText (latent)	**0.7436**	**0.7795**	0.2842	**6.7721**	-	-	-

Table 2. **Metrics comparison:** We evaluated our method against the baselines using automatic metrics (left) and human ratings (right). The results for the human ratings (right) are reported as the percentage of the majority vote raters that preferred our latent-based variant of our method over the baseline. As we can see, MAS (rand-label) achieves inferior results compared to the standard version of MAS, in both the automatic metrics and the user study.

vide an elaborate global prompt in addition to the local ones, i.e., to not require the user to repeat the same information twice. Hence, in order to reduce the domain gap between the training data and the input at inference time, we perform the following simple trick: we concatenate the local prompts to the global prompt at inference time separated by commas. Figure 16 demonstrates that this concatenation yields images that correspond to the local prompts better.

D. Additional Related Work

Image-to-image translation: Pix2Pix [16,43] utilized a conditional GAN [10, 25] to generate images from a paired image-segmentation dataset, which was later extended to the unpaired cased in CycleGAN [46]. UNIT [21] proposed to translate between domains using a shared latent space, which was extended to the multimodal [15] and few-shot [22] cases. SPADE [30] introduced spatially-adaptive normalization to achieve better results in segmentation-to-image task. However, all of these works, do not enable editing with a free-form text description.

Layout-to-image generation: The seminal paper of Reed *et al*. [36] generated images conditioned on location and attributes and managed to show controllability over single-instance images, but generating complex scenes was not demonstrated. Later works extended it to an entire layout [40–42, 45]. However, these methods do not support fine-grained control using free-form text prompts. Other methods [11, 12, 14, 19] proposed to condition the layout also on a global text, but they do not propose a fine-grained free-form control for each instance in the scene. In [32] an additional segmentation mask was introduced to control the shape of the instances in the scene, but they do not enable fine-grained free-form control for each instance separately. Recently [8] proposed to condition a GAN model on free-form captions and location bounding boxes, and showed promising results on synthetic datasets' generation, in contrast, we focus on fine-grained segmentation masks to control the shape (instead of coarse bounding boxes), and on generating natural images instead of synthetic ones.

Concurrently to our work, eDiff-I [4] presented a new text-to-image model that consists of an ensemble of expert denoising networks, each specializing in a specific noise interval. More related to our work, they proposed a training-free method, named paint-with-words, that enables users to specify the spatial locations of objects, by manipulating the cross-attention maps that correspond to the input tokens that they want to generate. Their method supports only rough segmentation maps, whereas our method focuses on the fine segmentation maps input case.

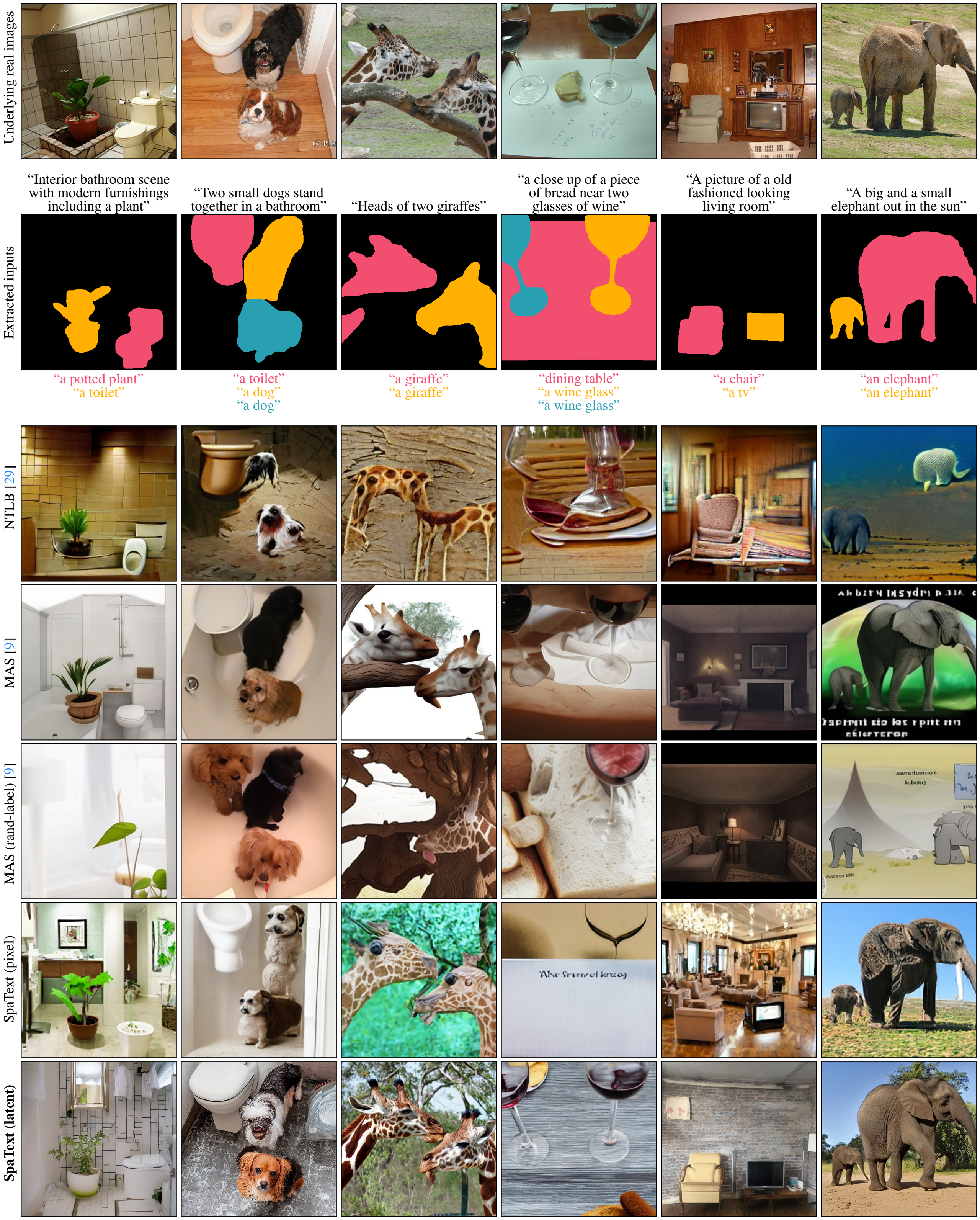

Figure 11. **Qualitative comparison on automatically generated inputs:** in order to create realistic inputs comparison, we utilized a segmentation dataset [20] to create inputs (second row) that are based on real images (top row). Given those inputs, we generate images using the baselines and the two variants of our method. As we can see, our latent-based variant of our method outperforms the baselines in terms of compliance with both the global and local texts, and in overall image quality.

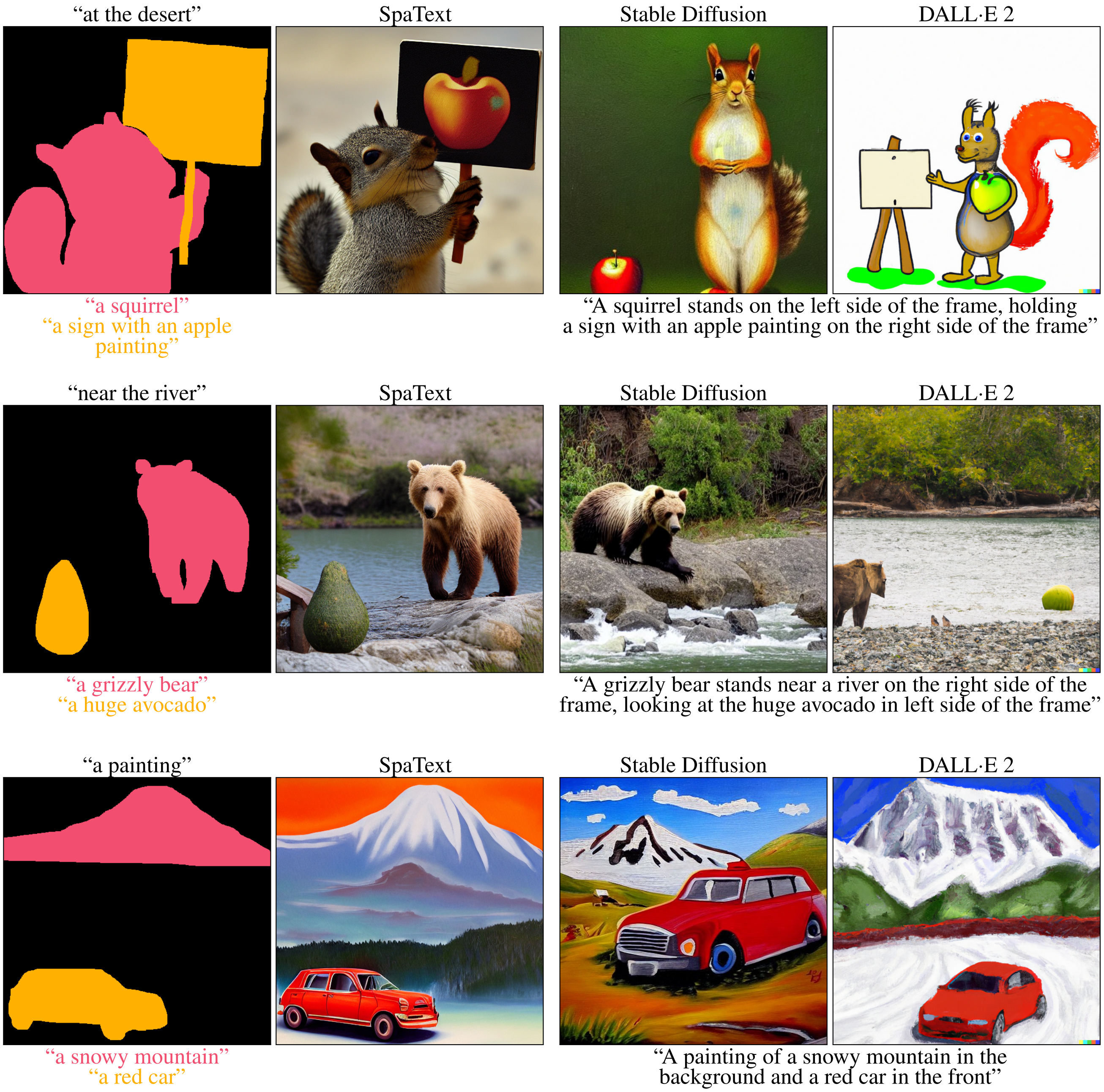
“at the desert”
SpaText
Stable Diffusion
DALL·E 2
“a squirrel”
“a sign with an apple painting”
“A squirrel stands on the left side of the frame, holding a sign with an apple painting on the right side of the frame”
“near the river”
SpaText
Stable Diffusion
DALL·E 2
“a grizzly bear”
“a huge avocado”
“A grizzly bear stands near a river on the right side of the frame, looking at the huge avocado in left side of the frame”
“a painting”
SpaText
Stable Diffusion
DALL·E 2
“a snowy mountain”
“a red car”
“A painting of a snowy mountain in the background and a red car in the front”

Figure 12. **Lack of fine-grained spatial control:** A user with a specific mental image (left) can easily generate it with a SpaText representation but will struggle to do so with traditional text-to-image models (right) [35, 37].

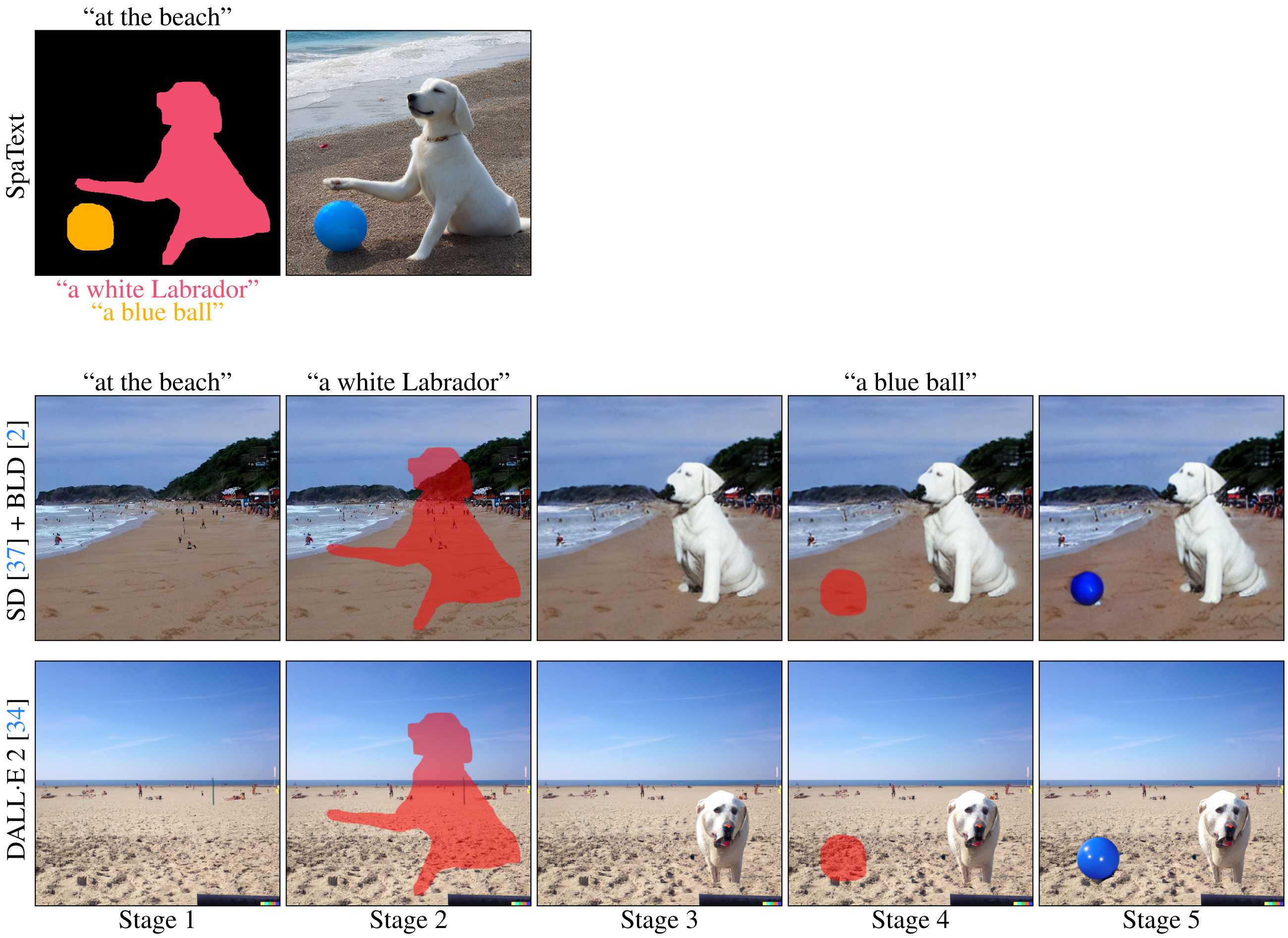

Figure 13. **Interactive editing baseline:** An alternative way to achieve image generation with free-form textual scene control as in our method (first row) is by iterative editing: at the first stage, the user can utilize a text-to-image model to generate the background of the scene, e.g. Stable Diffusion (second row) or DALL·E 2 (third row). Then, the user can sequentially mask the desired areas and provide the local prompts using a local text-driven editing method, e.g. Blended Latent Diffusion (second row) or DALL·E 2 (third row).

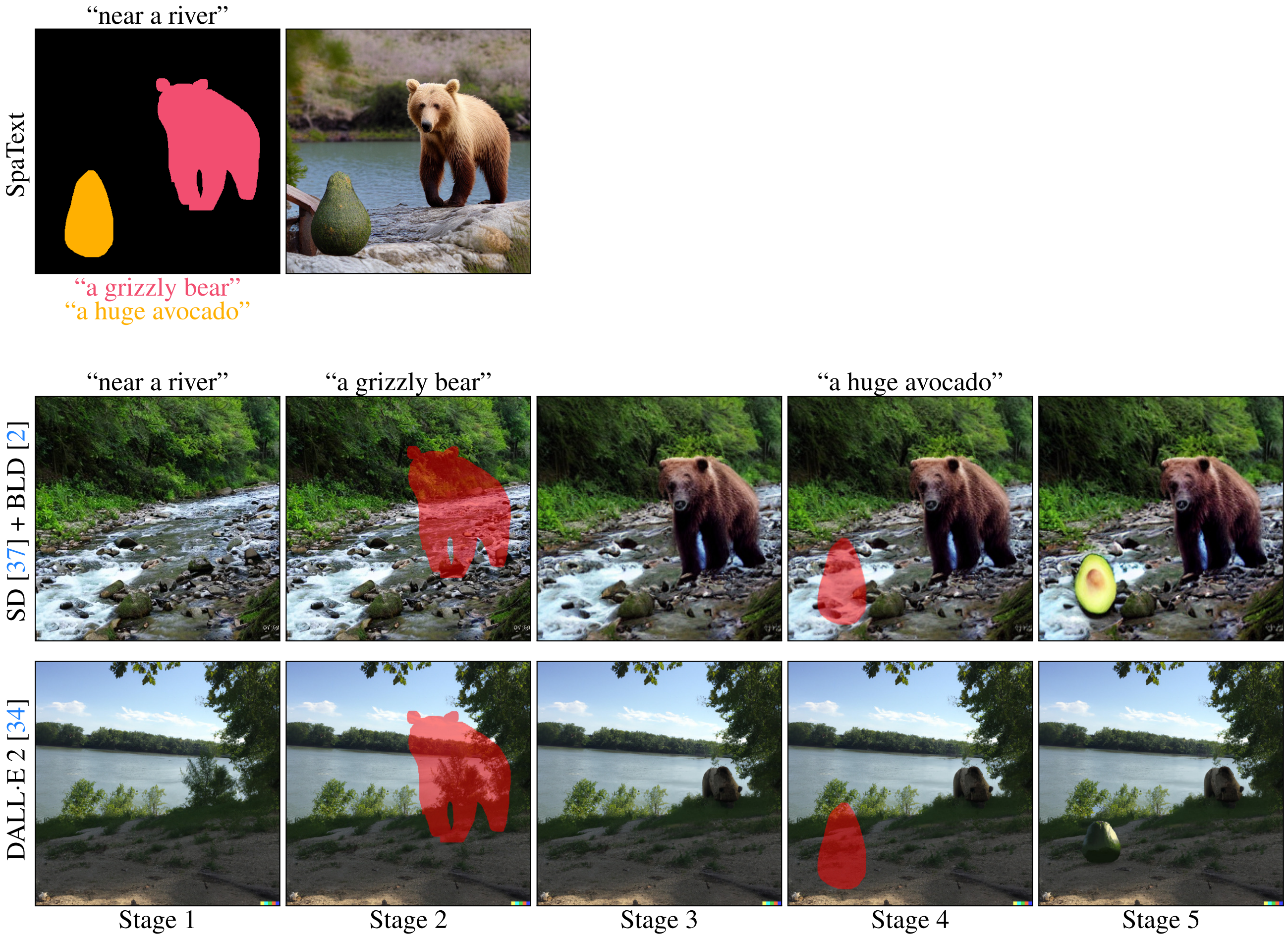

Figure 14. **Interactive editing baseline:** An alternative way to achieve image generation with free-form textual scene control as in our method (first row) is by iterative editing: at the first stage, the user can utilize a text-to-image model to generate the background of the scene, e.g. Stable Diffusion (second row) or DALL·E 2 (third row). Then, the user can sequentially mask the desired areas and provide the local prompts using a local text-driven editing method, e.g. Blended Latent Diffusion (second row) or DALL·E 2 (third row).

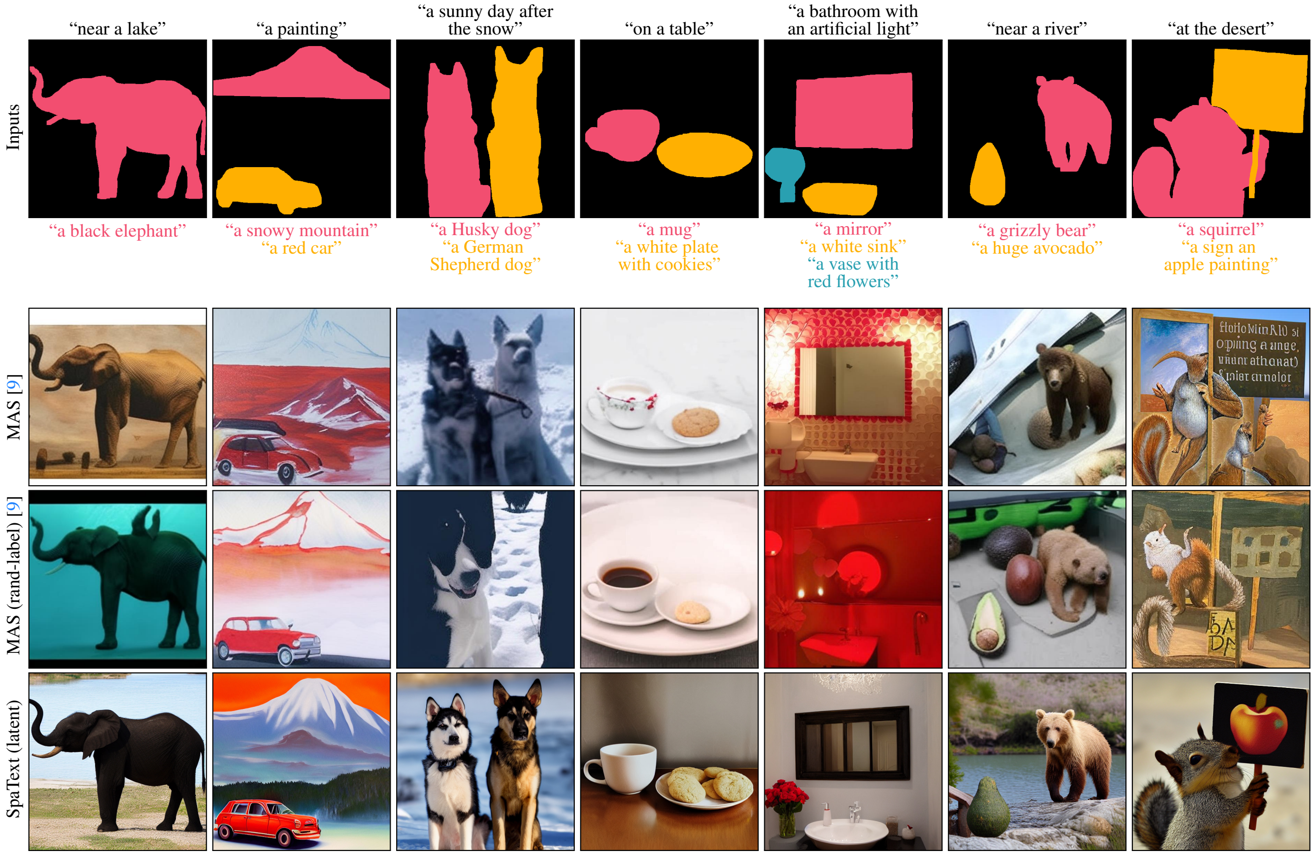

Figure 15. **Qualitative comparison of Make-A-Scene variants:** Given the inputs (top row), we generate images using the two variants of Make-A-Scene (adapted to our task as described in Appendix C.2) and our latent-based method. As we can see, SpaText (latent) outperforms these baselines in terms of compliance with both the global and local texts, and in overall image quality.

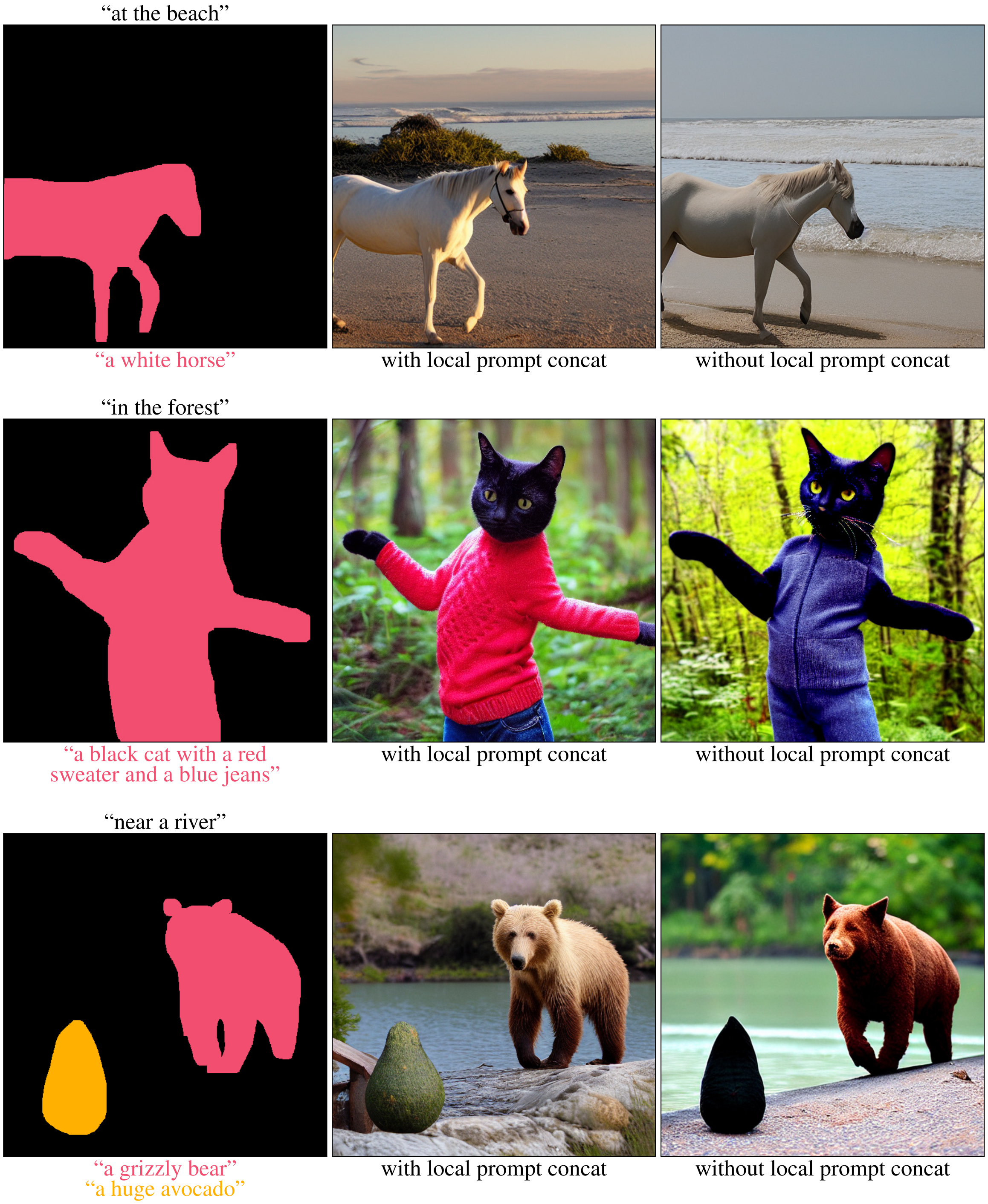

Figure 16. **Local prompts concatenation:** concatenating the local text prompts to the global prompt during inference mitigates the train-inference gap and enables better alignment between the generated images and the local prompts.

References

[1] Apple. No token left behind github. https://github.com/apple/ml-no-token-left-behind, 2022. 4

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022. 6, 14, 15

[3] 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. 6

[4] 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. 11

[5] CompVis. Stable diffusion github implementation. https://github.com/CompVis/stable-diffusion, 2022. 2

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3

[7] Hugging Face. Stable diffusion hugging face weights. https://huggingface.co/CompVis, 2022. 2

[8] Stanislav Frolov, Prateek Bansal, Jörn Hees, and Andreas Dengel. Dt2i: Dense text-to-image generation from region descriptions. *arXiv preprint arXiv:2204.02035*, 2022. 11

[9] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. *arXiv preprint arXiv:2203.13131*, 2022. 1, 4, 6, 9, 10, 11, 12, 16

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 11

[11] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Generating multiple objects at spatially distinct locations. In *International Conference on Learning Representations*, 2018. 11

[12] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 11

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. NeurIPS*, 2020. 1

[14] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7986–7994, 2018. 11

[15] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018. 11

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 11

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1, 2

[18] Bruno Klopfer and Douglas M Kelley. The rorschach technique. 1942. 8

[19] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12174–12182, 2019. 11

[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 *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 6, 12

[21] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30, 2017. 11

[22] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10551–10560, 2019. 11

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[24] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 9

[25] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 11

[26] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proc. ICML*, pages 8162–8171, 2021. 1

[27] OpenAI. DALL·E 2. https://github.com/openai/CLIP, 2021. 3

[28] OpenAI. Dalle2 demo. https://labs.openai.com/, 2022. 5

[29] Roni Paiss, Hila Chefer, and Lior Wolf. No token left behind: Explainability-aided image classification and generation. *arXiv preprint arXiv:2204.04908*, 2022. 4, 9, 10, 12

[30] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019. 11

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming

Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 4

[32] Dario Pavllo, Aurelien Lucchi, and Thomas Hofmann. Controlling style and semantics in weakly-supervised image generation. In *European conference on computer vision*, pages 482–499. Springer, 2020. 11

[33] 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. 3

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022. 1, 6, 14, 15

[35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 13

[36] Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. *Advances in neural information processing systems*, 29, 2016. 11

[37] 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, 2, 13, 14, 15

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 1, 3

[39] StabilityAI. Stable diffusion stabilityai demo. https://huggingface.co/spaces/stabilityai/stable-diffusion, 2022. 5

[40] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10531–10540, 2019. 11

[41] Wei Sun and Tianfu Wu. Learning layout and style reconfigurable gans for controllable image synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5070–5087, 2021. 11

[42] Tristan Sylvain, Pengchuan Zhang, Yoshua Bengio, R Devon Hjelm, and Shikhar Sharma. Object-centric image generation from layouts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2647–2655, 2021. 11

[43] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 11

[44] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. 2019. 3

[45] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8584–8593, 2019. 11

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 11