Compass Control: Multi-Object Orientation Control for Text-to-Image Generation

Supplementary Material

Appendix

Contents

A. Project page

Please load the Project website for interactive visualizations.

B. Controlling 3D orientation

The main text primarily focused on an orientation control for a single angle. However, our method is not limited to single orientation control, and we present an experiment for controlling all three orientation angles in a single model. Specifically, we updated the pose injection network to take 3 orientation angles as input to predict the pose token. We trained the model on flying objects - airplanes and helicopters- as rotation along all three axes is plausible for these objects. Specifically, we used six 3D assets from the web for these categories and followed the procedure in sec. 3.1 (main paper) to render the dataset. We present results for controlling all the 3 orientation angles in Fig. 1 and 2. In Fig. 1, we present rotation along all three axes for a fighter jet aircraft in three separate rows. Observe that, our method can precisely control all the object orientations along all the three axis. In Fig. 2, we show the generalization of our trained model in controlling the orientation of a variety of objects. Notably, our model is not trained on birds or rockets. Still, it can generate consistent orientation-conditioned scenes following the text prompts. Note, that the compass shown in the figure is just for visualization purposes (can have an error of a few degrees).

C. Additional Control

Continuous control for camera elevation. Our proposed conditioning mechanism is generalized and can be adapted to achieve continuous camera elevation control in Fig. 3. We generated a dataset with camera elevation variations and conditioned the denoising UNet on elevation angle.

Control for object scale. We can also precisely control the size of individual objects with additional conditioning on the object scale, as shown in Fig. 3. Specifically, we condition the diffusion model with the length of the diagonal of a tight 2D bounding box.

D. Generalization to StableDiffusion-XL

We have presented all the results on StableDiffusion-2.1 in the main paper. Our method also generalizes well to a larger StableDiffusion-XL backbone model shown in Fig. 4. The results demonstrate improved image quality with accurate orientation control of the generated objects.

E. Diverse poses for non-rigid objects

We build our dataset with only a few synthetic objects in their fixed canonical pose to generate our training data. This makes the model prone to overfitting on these poses for the non-rigid objects in the dataset - dog, horse, and lion. For instance, during inference, the model can generate only standing dogs in the given orientation. We generate augmentations with realistic pose variations in the training data to mitigate this. Specifically, we randomly mask some regions from the Canny Edge map and pass it to the ControlNet (Fig. 5a). This allows ControlNet to freely generate any plausible pose within the masked region. When trained with

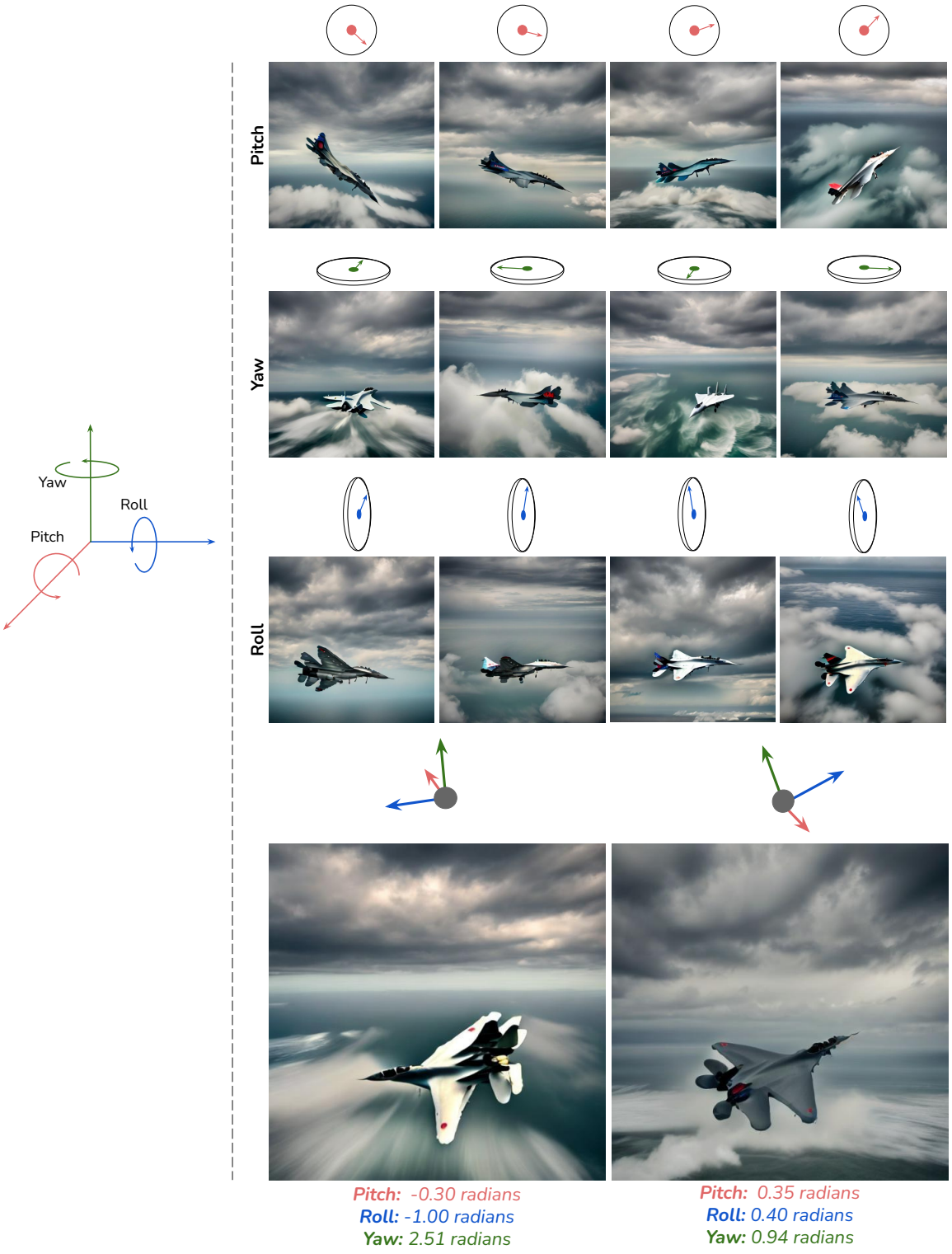


Figure 1. **Conditioning on all three orientation angles for a single object.**

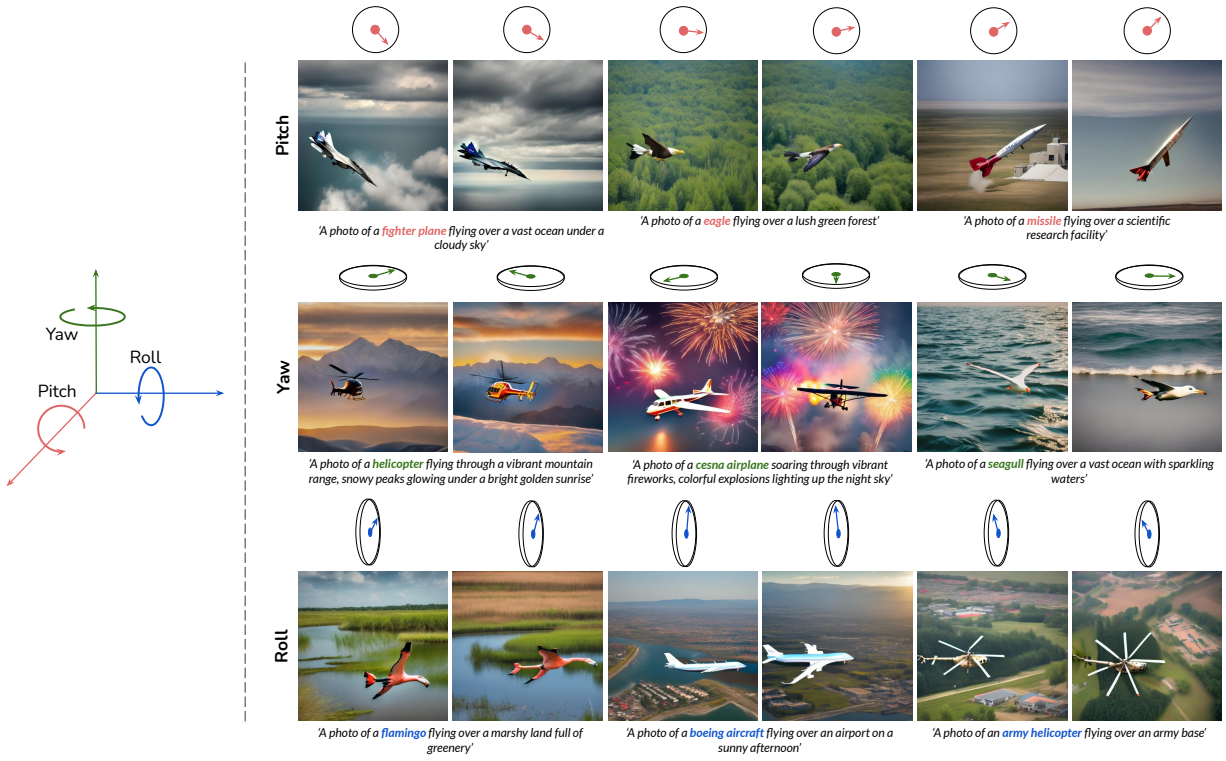


Figure 2. **Conditioning on all three orientation angles**

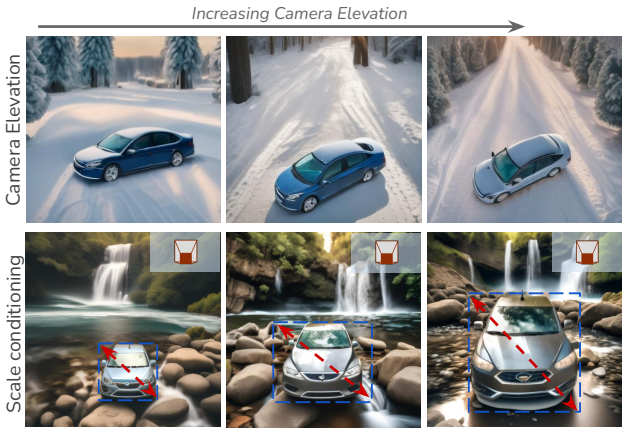


Figure 3. Additional Controls

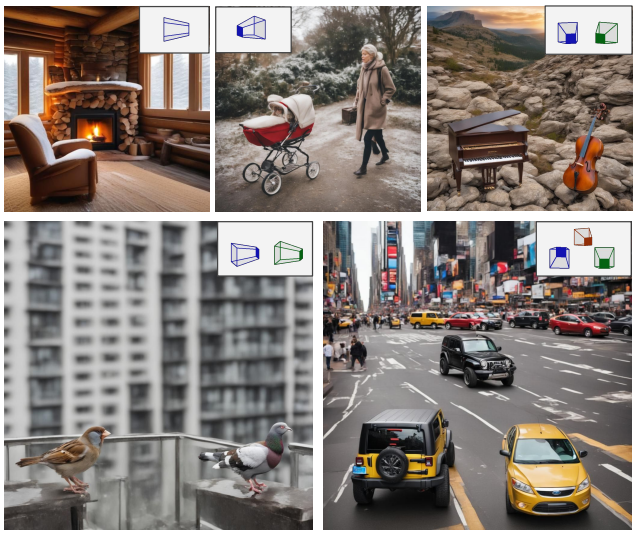


Figure 4. Compass Control on StableDiffusion-XL.

resulting augmentations, our method can generate diverse pose variations of non-rigid training objects while following the precise orientation as shown in Fig. 5b).

F. Robustness to the 2D bounding boxes

Coarse bounding boxes. We analyze the robustness of required 2D object bounding boxes during the inference. First, we analyze the effect of the coarseness of the bounding box on the generated scenes in Fig. 6. Our model does not generate objects that tightly occupy the provided bounding box. This is convenient for the user, as they don't have to provide an exact 2D bounding box. We present results for different bounding box sizes while keeping the center fixed.

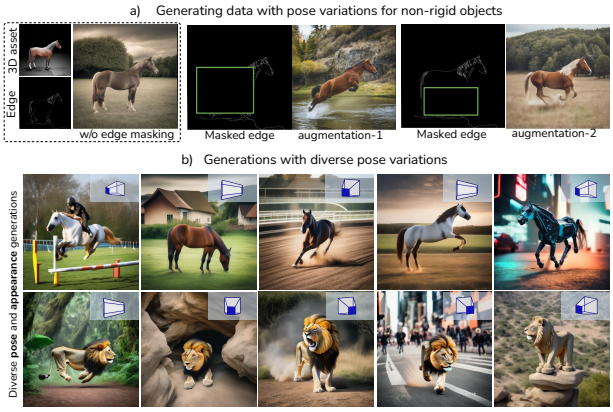


Figure 5. Pose variations for non-rigid objects

The model is robust to size changes and generates realistic scene compositions. The objects fall inside the box but they don't tightly fit the box. This provides more flexibility to the base generative model in generating more realistic scenes with relaxed constraints than conditioning on precise bounding boxes.

Spawning random boxes. In another experiment, we randomly spawn non-overlapping boxes, eliminating the user requirement to provide 2D boxes. The results are present in Fig. 6. Our method generates realistic compositions for these random layouts, with precise orientation control. The proposed design of using *loose* bounding boxes during training, enables this, as the objects can adjust their size within the box region to make coherent scenes.

Overlapping boxes. We present an ablation with the amount of overlap of 2D object boxes in Fig. 7 during inference. Our method can handle the overlap between 2D boxes upto a good extent. On increasing the overlap the models' performance gracefully degrades in the pose control as the overlapping region is controlled by both the pose tokens (jeep in 4th example). With a large overlap in the bounding boxes, the model fails to generate both objects, and this is one of the limitations of our proposed approach, which is based on attention regularization. However, this limitation is common across all the bounding box conditioned or guided generative models.

G. Discussion with SoTA object-centric works.

We compare the framework of our approach with recent works on object-centric 3D control in generation and editing with diffusion models. Particularly, we contrast our method with Neural Assets [10] and LooseControl [1], as these two are the closest method to ours. We present a comparison with both these methods at an approach level in Tab. 1.

H. Synthetic data generation

We render scenes with 3D assets in a Blender [5] environment for our dataset. Specifically, we place an opaque floor on the $x - y$ plane and place a camera tilted slightly towards the ground at a fixed position. The scene is lighted using 3 point lights of random intensity, placed at random locations. Once the environment is ready, we place the 3D assets at random locations and orientations and render the scene. For each rendered image we store the identity of the 3D assets in it, their respective orientations and 2D bounding boxes. We constrain the locations and orientations so that the object completely lies within the rendered image. Additionally, for two object scenes, we ensure that their 2D bounding boxes do not overlap. In all, we have 1000 one-object scenes and 7900 two-object scenes. Some samples from the rendered images can be found in Fig. 8.
However, training on this dataset alone leads to over-fitting to the plain backgrounds, as we have presented in the ablative experiments in the main text. Therefore, to generate the objects in diverse contexts, we augment the rendered scenes using Canny ControlNet [11]. Specifically, given a rendered scene, we extract it's Canny map using OpenCV [2], with the low and high thresholds set to 100 and 200 respectively. We use the following prompts for the augmentations:

1. a photo of $\langle subject \rangle$ in a snowy forest, with a gentle snowfall and snow-covered trees
2. a photo of $\langle subject \rangle$ in a vast desert with towering sand dunes and a clear blue sky
3. a photo of $\langle subject \rangle$ in a medieval castle courtyard with ancient stone walls and archways
4. a photo of $\langle subject \rangle$ in a sunflower field under a clear blue sky
5. a photo of $\langle subject \rangle$ in a dense rainforest, with sunlight streaming through the canopy
6. a photo of $\langle subject \rangle$ in a serene Japanese garden, surrounded by cherry blossoms
7. a photo of $\langle subject \rangle$ on a rocky cliff overlooking a vast ocean
8. a photo of $\langle subject \rangle$ by a riverside with wildflowers blooming nearby
9. a photo of $\langle subject \rangle$ at a river's edge with stones scattered around
10. a photo of $\langle subject \rangle$ in front of the Eiffel Tower at sunset
11. a photo of $\langle subject \rangle$ in a vibrant autumn forest, with orange and red leaves carpeting the ground
12. a photo of $\langle subject \rangle$ in a vast open plain, with golden grasses swaying in the wind and distant mountains on the horizon under a wide, clear sky
13. a photo of $\langle subject \rangle$ on a cobblestone street in a quaint European village, with flower-filled balconies and historic buildings
14. a photo of $\langle subject \rangle$ in a canyon with towering red rock

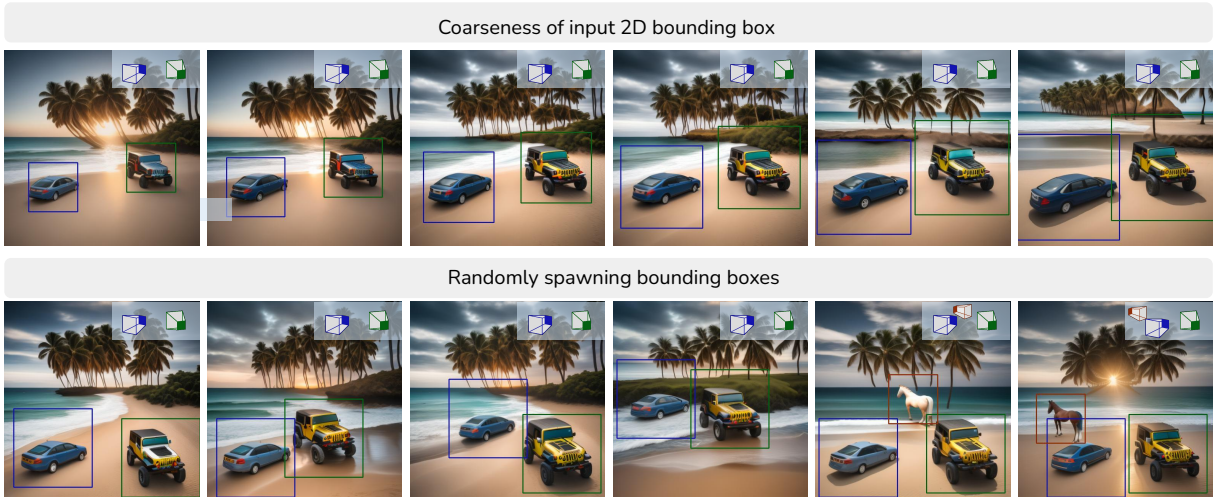


Figure 6. **Robustness of 2D bounding boxes.** Our method generates realistic scene compositions with different 2D bounding box sizes. Allowing for a loose bounding box during training provides this flexibility to the model to generate realistic scenes while coarsely following the input 2D box. Further, random non-overlapping boxes can also be spawned during inference without any degradation in quality. This robustness to the actual bounding box shape, reduces the burden on the user and is enabled by the *loose* bounding box used during training.

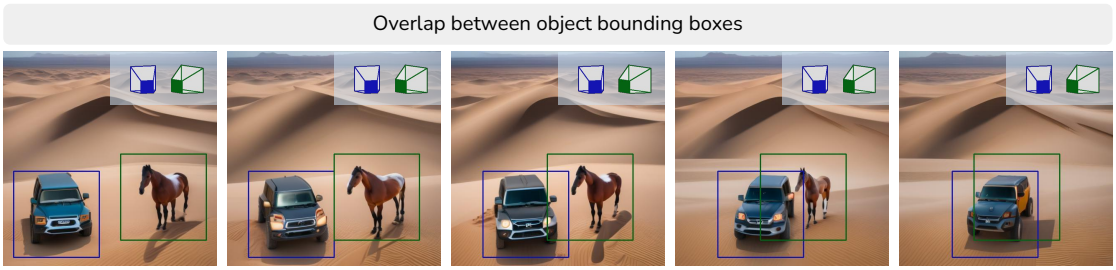


Figure 7. **Overlapping bounding boxes.** Our method can handle overlap between the two input bounding boxes up to a good extent. However, with a large overlap, the model struggles to generate accurate orientations (jeep in the fourth example), due to the mixing of pose tokens.

formations, and scattered desert plants growing in the rocky terrain

We run this augmentation pipeline on all the rendered images, and do a manual filtering to remove the inconsistent generations. In all, we have 771 single-object augmentations and 5239 two-object augmentations.

I. Orientation Regressor

We train a neural network model to predict the orientation angle of an object in the generated image. We use a pre-trained ResNet-18 [6] as the feature extractor and a mlp head consisting of two hidden layers of 128 neurons, each with ReLU activations. Finally, we predict a single orientation angle θ along the up-axis (details in the main text - sec.3.1). We call this model *orientation regressor* and train with a dataset of $35K$ images generated by rendering 30 synthetic 3D assets of the test object categories followed by their canny ControlNet augmentations. This data is highly diverse, containing various backgrounds and object appearances, enabling the learning of an accurate orientation regressor. We train with a batch size of 128, a learning rate of $5e-5$ for 95 epochs with Adam optimizer. On an unseen test set of $8K$ images from the same distribution, the trained model achieves a mean angular error of 0.125. Further, we present the results for evaluation on a completely unseen dataset, generated by Stable Diffusion [9], containing the test objects in Fig. 9. We can observe that the trained orientation regressor predicts accurate orientations, and hence, it is a good estimator for evaluating pose consistency. In the case of multi-object scenes, we crop out the objects using Grounding DINO [8] and pass them to the *orientation regressor*.

	Model type	Training data	Input during inference	Novel categories	Input Representation	Personalization
LooseControl [1]	Generation	Real images (w 3D boxes)	3D object boxes	Yes	Explicit 3D (Depth)	No
Neural Assets [10]	Editing	Real videos (w 3D boxes)	3D object boxes	No	Implicit (List of bbox)	Yes
Ours	Generation	Synthetic images (w Orientation + 2D boxes)	Orientation + 2D object boxes	Yes	Implicit List of orientations	Yes

Table 1. Comparison with state-of-the-art approaches for object-centric control in the generation process.

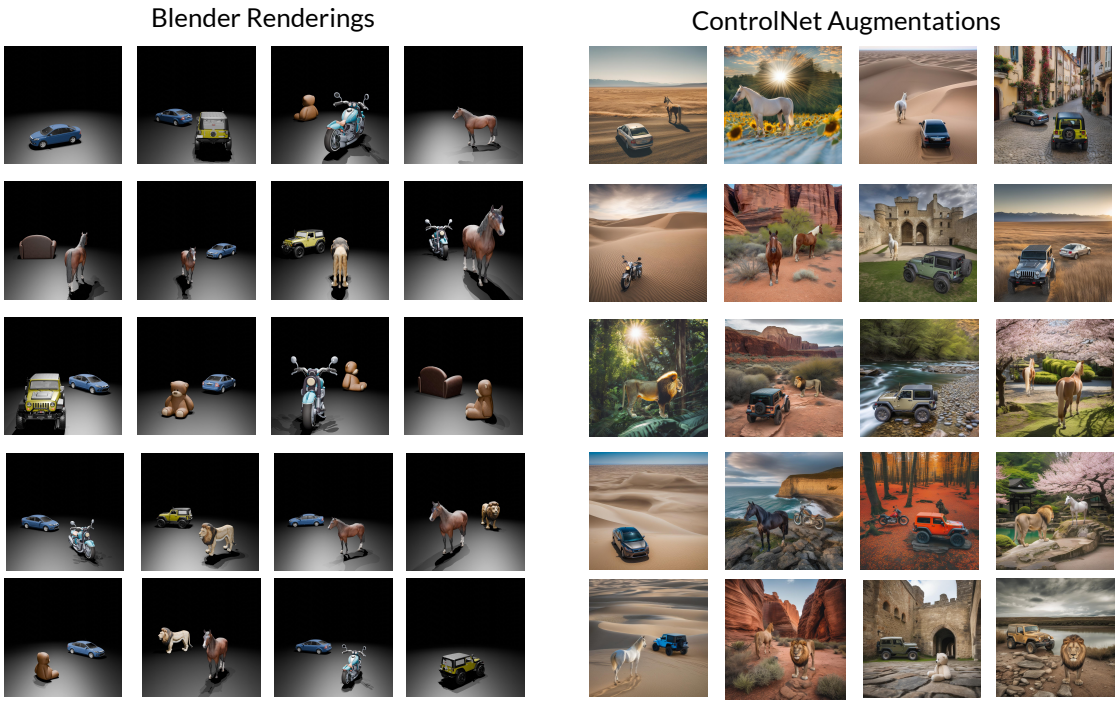


Figure 8. Samples from data generation process

J. Baseline details

We provide implementation details for the baselines discussed in the paper.

J.1. ViewNeTI [3]

ViewNeTI trains a small MLP to project the 3D camera pose to 3D view token. This token, along with the text prompt, is used to condition the text-to-image model. In the basic form, it is trained on a single scene with multi-view images and 3D camera poses. Once trained, the model can generate novel views for the trained scene. However, in an extended version, it is trained with multiple scenes to learn a generalizable view token. This token is then used for view control in text-to-image generation. For comparison, we use this version and train on our synthetic dataset of rendered multi-view scenes. Specifically, instead of conditioning on 3D camera pose, we condition on orientation angle θ and predict the view token. We train the model for $60K$ iterations on 1000 multi-view images of 10 assets. Note that because this model only supports a global view control, we train and evaluate it on only single object scenes for orientation control.

J.2. Continous 3D Words [4]

In this approach, a text-to-image diffusion model is conditioned on continuous 3D tokens to control 3D attributes such as lighting and object pose. They learn a generalizable *3D word* in the text embedding space of the T2I model for each attribute, which is used along with the text prompt for conditioning. To learn the 3D word token, they use renderings of a single object and generate its augmentations with depth-conditioned ControlNet. However, it is essential that the 3D word token is disentangled from the object used for training. For this, they follow a staged training procedure: first learn the object's appearance (stage 1), and then learn the 3D attribute (stage 2). Following this, we train this

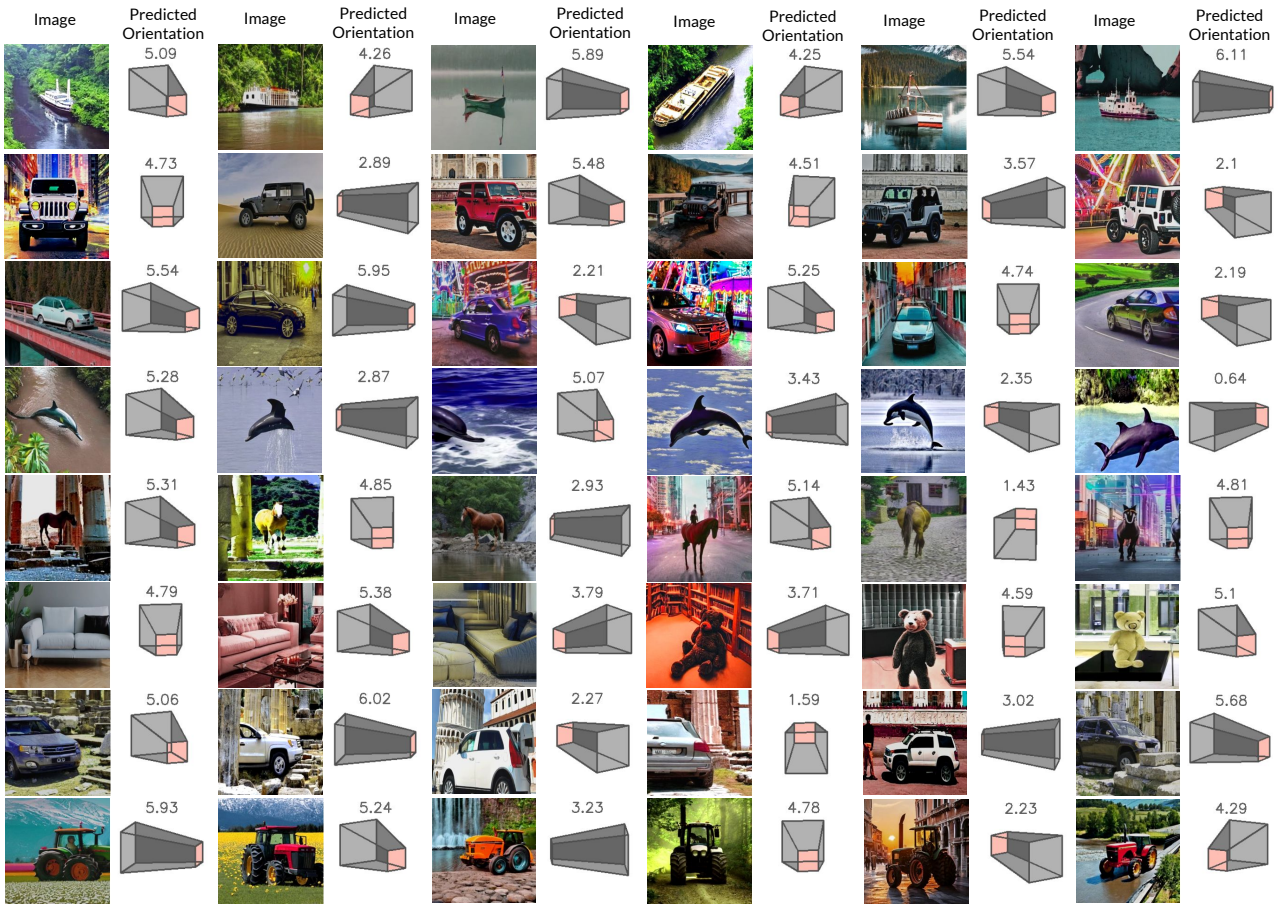


Figure 9. Predictions of the trained orientation regressor on unseen samples generated from Stable Diffusion [9]. The model can predict the orientations accurately for the diverse unseen data and acts as a good critic to evaluate orientation consistency in generated images.

model a single 3D asset, *sedan*. We train for 5000 iterations in stage 1 and 15000 iterations in stage 2 (same as the original model). However, the trained model poorly generalizes to new objects as it is trained on a single object mesh (Fig.7 in the main text).

Here we present a variant of this model, which is trained on multiple 3D assets instead of just one (as proposed in their original paper). We use the same rendered images dataset as ours, and augment it using their proposed augmentation strategy. Notably, this dataset has diverse layouts and objects placed at random locations in the scene, making the learning process challenging. Since this model only allows for global control, we train and evaluate it on single object scenes only. We perform 30000 training iterations in the first stage to learn the object appearance, followed by 70000 iterations to learn the 3D word token. The comparison is presented in Fig. 10. Our method achieves superior performance as compared to this baseline. The baseline struggles in pose control due to high diversity in the scene layouts, highlighting the importance of our attention localization mechanism CALL. Further, our backgrounds are much richer, as we use canny-conditioned ControlNet augmentations, which leads to richer augmentations.

J.3. LooseControl [1]

LooseControl is a conditioning framework on text-to-image diffusion models that allows for 3D scene layout control. The framework is built on a depth-conditioned ControlNet model. However, instead of relying on accurate depth maps, which are often difficult to construct, LooseControl conditions the generation on coarse depth maps. Specifically, in this loose depth map, the scene boundaries are represented as planes, and the objects are represented by their loose 3D bounding boxes. LooseControl is implemented as a LoRA [7] fine-tuning over depth-conditioned ControlNet model. This fine-tuning enables it to condition the generation using loose object depth maps also, against the accurate depth maps required by original ControlNet. In our exper-

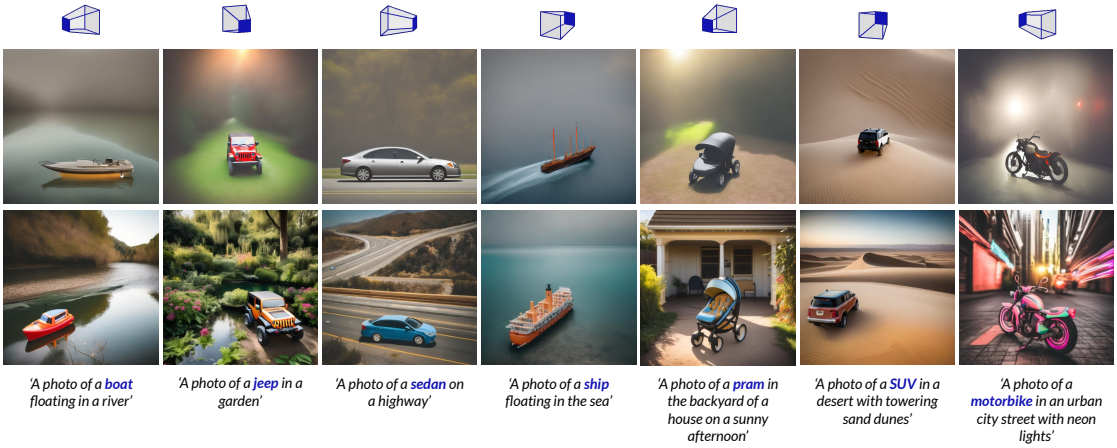


Figure 10. Comparsison with modified Continuous 3D Words [4] trained on multiple assets. Compass control generates more realistic outputs and follows the text prompt better than the Cont-3D-Words trained on multiple object datasets.

iments, we generate the loose depth maps by placing 3D bounding boxes in a Blender [5] environment, and rendering the depth from camera viewpoint. Specifically, we randomly sample objects' locations and pose within the scene boundary and place a 3D bounding box for each object. Notably, one can control the object orientation by rotating the corresponding 3D bounding box in the input. We define a fixed template of 3D bounding box dimensions for each test object in the dataset. The obtained depth maps are used to condition the model. We used the publicly available checkpoint for LooseControl in our evaluation. As this method allows for multi-object control, we compare both single and multi-object scenes. However, in experiments, we observe that LooseControl struggles to generate multi-object scenes with precise pose control and often resorts to generating bounding box artifacts. This is primarily due to the strong depth conditioning prior in the base depth ControlNet model, which is trained to follow exact depth maps.

K. Additional Results

K.1. Comparisons

We present additional baseline comparison results in Fig. 12. Our method follows the text prompts and generates objects following the input prompts

L. Implementation Details

L.1. Method details

We use Stable Diffusion v2.1 [9] as our base T2I model and use LoRA rank 4 for fine-tuning its UNet. Our encoder model $\mathcal{P}$ is a lightweight MLP: three linear layers with ReLU. We train our model for $100K$ iterations with a batch size of 4 with AdamW optimizer and a fixed learning rate of 10^{-4}. We train first stage for $30K$ iterations with only single object scenes and the next stage for $70K$ iterations with mix of single and two subject scenes. We use SD-Xl for generating augmentations due to its higher realism.

We keep the bounding box padding $\lambda = 1.2$ for CALL. The training takes 24 hours on a single A6000 GPU, thus highly efficient.

L.2. Evaluation dataset

We randomly sample 10 pose orientation in the range of $(0{,}360\,\mathrm{deg})$ for each prompt and object combination. We used the following set of prompts for evaluation, containing single and two subject. In each prompt *¡subject¿* is replaced with a single subject (e.g., jeep) or two subjects (e.g., jeep and sedan). Notably these prompts are different that the one used to generate ControlNet augmentations, to accurately evaluate model generalization.

For road objects

1. A photo of $\langle subject \rangle$ in front of the Taj Mahal
2. A photo of $\langle subject \rangle$ on the streets of Venice, with the sun setting in the background
3. A photo of $\langle subject \rangle$ in front of the leaning tower of Pisa in Italy
4. A photo of $\langle subject \rangle$ in a modern city street surrounded by towering skyscrapers and neon lights
5. A photo of $\langle subject \rangle$ in an ancient Greek temple ruin, with broken columns and weathered stone steps
6. A photo of $\langle subject \rangle$ in a field of dandelions, with snowy mountain peaks in the distance
7. A photo of $\langle subject \rangle$ in a rustic village with cobblestone streets and small houses
8. A photo of $\langle subject \rangle$ on a winding country road with green fields, trees, and distant mountains under a sunny sky
9. A photo of $\langle subject \rangle$ in front of a serene waterfall with

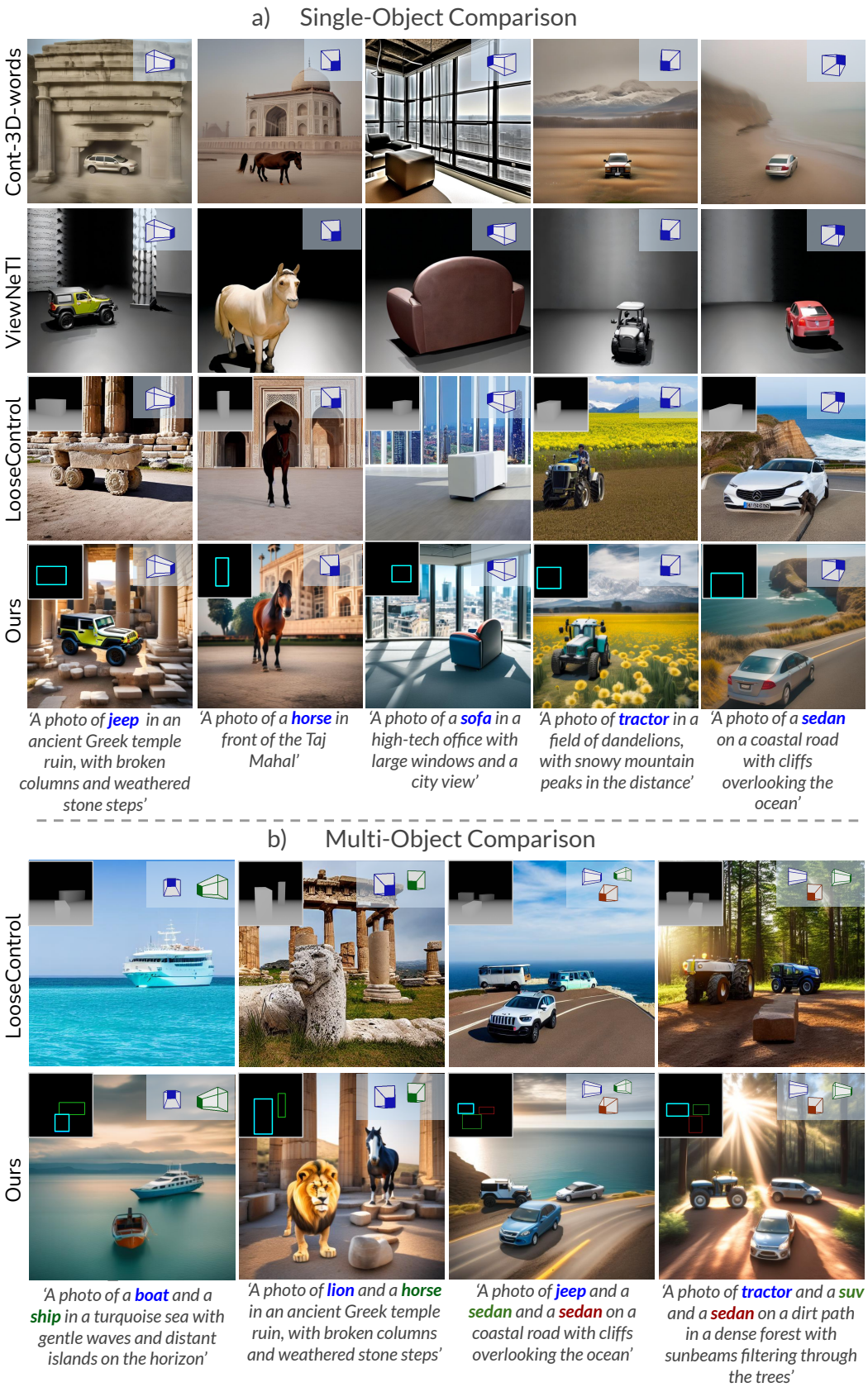
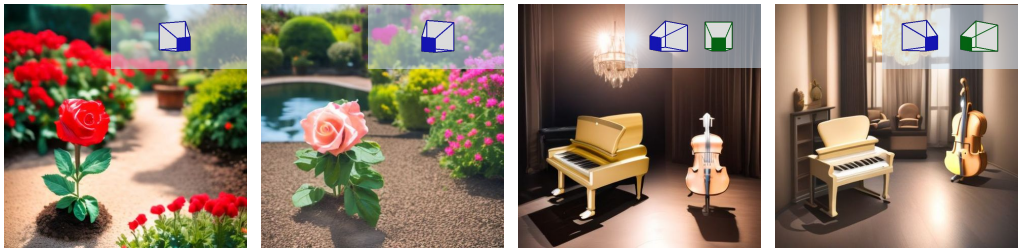
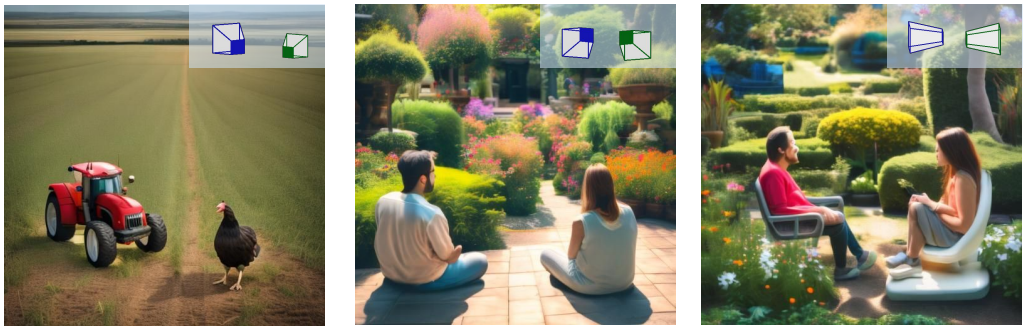


Figure 11. Additional comparison results with the baselines for single object and multi-object scenes.



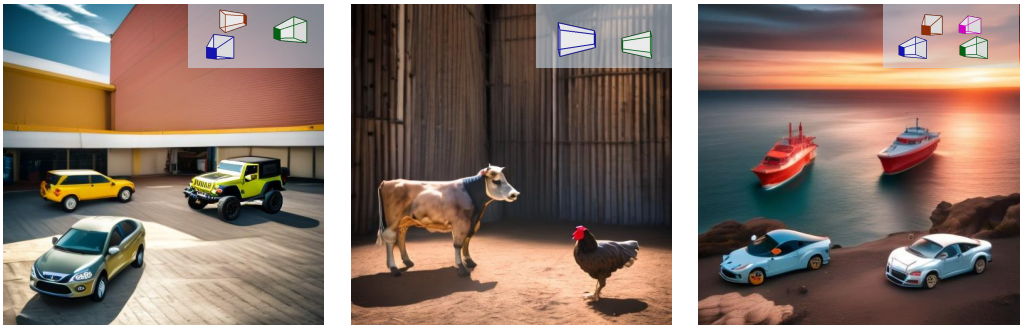
'A photo of a **rose** flower in a beautiful garden'

'A photo of a **piano** and a **cello** in a modern living room with soft yellow lighting from the chandeliar'



'A photo of a **tractor** and a **hen** in a farm'

'A **man** and a **woman** talking to each other sitting in a garden'



'A photo of a **sedan** and a **SUV** and a **jeep** in the parking of a mall'

'A photo of a **cow** and a **hen** in a barn'

'A photo of a **Ferrari** and a **Bugatti** racing furiously on a winding coastal road under a fiery sunset, cliffs, **ship** and **ship** is floating on the ocean in view'

Figure 12. More qualitative results from our method, *Compass Control*.

trees scattered around the region, and stones scattered in the water

10. A photo of $\langle subject \rangle$ on a sandy desert road with dunes and a vast, open sky above
11. A photo of $\langle subject \rangle$ on a bridge overlooking a river with mountains in the background
12. A photo of $\langle subject \rangle$ on a dirt path in a dense forest with sunbeams filtering through the trees
13. A photo of $\langle subject \rangle$ on a coastal road with cliffs overlooking the ocean

14. A photo of $\langle subject \rangle$ in front of a historical castle with high stone walls and flags flying in the breeze
15. A photo of $\langle subject \rangle$ in front of an amusement park with bright lights and ferris wheels in the background

For water objects

1. A photo of $\langle subject \rangle$ on still waters under a cloudy sky, mountains visible in the distant horizon
2. A photo of $\langle subject \rangle$ floating on a misty lake, surrounded by calm waters and serene, foggy atmosphere
3. A photo of $\langle subject \rangle$ in the vast sea, with a clear blue sky and a few fluffy clouds
4. A photo of $\langle subject \rangle$ in the middle of a stormy ocean, with dark clouds and crashing waves
5. A photo of $\langle subject \rangle$ in a calm lake with lily pads and reeds growing near the shoreline
6. A photo of $\langle subject \rangle$ on a river running through a dense jungle with vibrant green foliage
7. A photo of $\langle subject \rangle$ in a mountain lake surrounded by pine trees and snow-capped peaks
8. A photo of $\langle subject \rangle$ floating in a lagoon with tropical fish and coral visible beneath the water
9. A photo of $\langle subject \rangle$ on a frozen lake with a snowy landscape surrounding it
10. A photo of $\langle subject \rangle$ on a serene river at dusk, with reflections of the sunset on the water
11. A photo of $\langle subject \rangle$ in the middle of a vast marshland with tall grasses and migratory birds flying overhead
12. A photo of $\langle subject \rangle$ near a small waterfall cascading into a clear pool in a rocky area
13. A photo of $\langle subject \rangle$ on a bay with large rock formations jutting out of the water
14. A photo of $\langle subject \rangle$ in a turquoise sea with gentle waves and distant islands on the horizon
15. A photo of $\langle subject \rangle$ in a narrow canal in an old European city, with historic buildings lining the waterway

For indoor objects

1. A photo of $\langle subject \rangle$ in a modern living room setting with painted walls and glass windows
2. A photo of $\langle subject \rangle$ in a minimalist living room
3. A photo of $\langle subject \rangle$ in a cozy library with shelves filled with books and warm lighting
4. A photo of $\langle subject \rangle$ in a high-tech office with large windows and a city view
5. A photo of $\langle subject \rangle$ in an art studio with canvas paintings and art supplies scattered around
6. A photo of $\langle subject \rangle$ in a rustic kitchen with wooden cabinets and a stone countertop
7. A photo of $\langle subject \rangle$ in a lavish living room with elegant decor and soft lighting
8. A photo of $\langle subject \rangle$ in a large dining hall with chandeliers and long tables
9. A photo of $\langle subject \rangle$ in a traditional Japanese tatami room with sliding paper doors
10. A photo of $\langle subject \rangle$ in a well-equipped gym with weights and fitness machines
11. A photo of $\langle subject \rangle$ in a music studio with soundproof walls and musical instruments
12. A photo of $\langle subject \rangle$ in a sunlit greenhouse filled with tropical plants
13. A photo of $\langle subject \rangle$ in a children's playroom with colorful toys and posters on the walls
14. A photo of $\langle subject \rangle$ in an underground wine cellar with wooden barrels and dim lighting
15. A photo of $\langle subject \rangle$ in a cozy reading nook with a soft armchair and a small lamp

References

[1] Shariq Farooq Bhat, Niloy Mitra, and Peter Wonka. Loosecontrol: Lifting controlnet for generalized depth conditioning. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 1, 4, 6, 7

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 4

[3] James Burgess, Kuan-Chieh Wang, and Serena Yeung. Viewpoint textual inversion: Unleashing novel view synthesis with pretrained 2d diffusion models. *arXiv preprint arXiv:2309.07986*, 2023. 1, 6

[4] Ta-Ying Cheng, Matheus Gadelha, Thibault Groueix, Matthew Fisher, Radomir Mech, Andrew Markham, and Niki Trigoni. Learning continuous 3d words for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6753–6762, 2024. 1, 6, 8

[5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 4, 8

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 7

[8] 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. 5

[9] 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. 5, 7, 8

[10] Ziyi Wu, Yulia Rubanova, Rishabh Kabra, Drew A Hudson, Igor Gilitschenski, Yusuf Aytar, Sjoerd van Steenkiste, Kelsey R Allen, and Thomas Kipf. Neural assets: 3d-aware multi-object scene synthesis with image diffusion models. *arXiv preprint arXiv:2406.09292*, 2024. 4, 6

[11] 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. 4