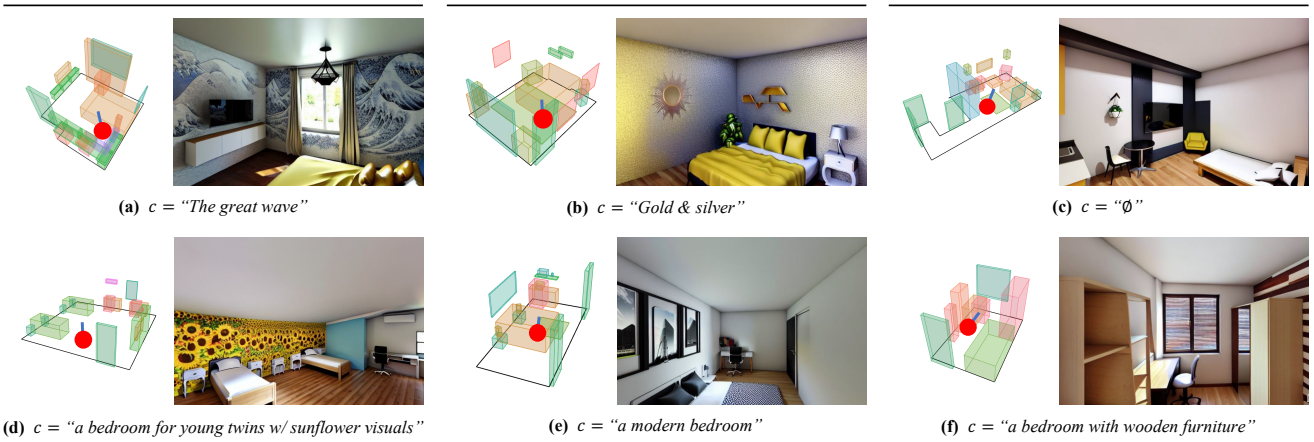
LACONIC: A 3D Layout Adapter for Controllable Image Creation

Supplementary Material



(a) c = *"The great wave"* **(b)** c = *"Gold & silver"* **(c)** c = *"∅"*

(d) c = *"a bedroom for young twins w/ sunflower visuals"* **(e)** c = *"a modern bedroom"* **(f)** c = *"a bedroom with wooden furniture"*

Figure 1. **Additional layout-guided image synthesis results from diverse prompts**. The generated images (Right) from the input 3D layouts (Left) and text prompts demonstrate our method's strong adherence to both conditions. Notably, LACONIC produces high-quality results across various settings: without a text prompt (c), with in-domain prompts (e, f), and with out-of-distribution prompts (a, b, d).

A. Implementation

In this section, we provide comprehensive implementation details of our 3D layout adapter architecture, pretrained text-to-image diffusion backbone, and our training and inference settings. Unless otherwise specified, the same settings are used for all datasets in our experiments. We also detail the training and test configurations of the baselines.

A.1. Network Architecture

A.1.1. Pretrained Models

We employ the established and widely adopted Stable Diffusion [21] v1.5 to implement the conditional UNet image denoiser ϵ_θ and CLIP [19] text prompt encoder τ_θ, using the pretrained weights and implementation from the HuggingFace Diffusers [28] library. In our framework, the text encoder is used to encode both the *global* image caption c and the *object-level* semantic descriptions s.

A.1.2. 3D Layout Encoder

We detail the modules used to embed the input semantic 3D layout $\mathcal{S}$, which build upon previous work [14, 15, 29] and are illustrated in Figure 3 and described in Section 3.3 in the main paper.

Shared Object Encoder Parametric object bounding box representations o_i are embedded by a common module. More precisely, each scalar defining individual object spatial attributes (p_i, R_i, d_i) is projected to a higher-dimensional vector using fixed, sinusoidal positional encoding [27] with 32 frequencies. We follow [29] and apply:

$$PE(k) = \{\sin(128^{j/31}k), \cos(128^{j/31}k)\}_{j=0}^{31} \in \mathbb{R}^{64} \quad (1)$$

Consequently, object's *position* p_i and *dimension* d_i are encoded to 192-dimensional attributes. Importantly, the *rotation* matrix R_i is first expressed in a continuous representation, following the recommendations from [34], leading to a vector in $\mathbb{R}^6$ from which individual scalars are also processed by Equation (1). This produces a representation in $\mathbb{R}^{384}$ that is subsequently mapped by a linear layer to $\mathbb{R}^{192}$. The semantic embedding $\tau(s_i)$ is the *end-of-sequence* output token from the pretrained text encoder in $\mathbb{R}^{d_\tau}$, with $d_\tau = 768$, and is further projected to a 192-dimensional representation by an MLP featuring a single hidden layer of dimension 384 with LeakyReLU activations. Encoded object attributes are finally concatenated, yielding an individual token $\mathcal{T}_{o_i}$ of dimension $4 \times 192 = d_\tau$ for each of the N objects in the scene.

Floor Plan Encoder Following prior work [14, 29], we encode the optional scene's floor plan $\mathcal{F}$ by leveraging a popular implementation of the PointNet [18] model, applied on a point cloud representation obtained by sampling $P = 100$ three-dimensional points that are evenly spaced along the room's boundaries. A single *floor* token $\mathcal{T}_{\mathcal{F}_i} \in \mathbb{R}^{d_\tau}$ is subsequently obtained by projecting the 1024-dimensional module's output.

Transformer Encoder New token representations $\hat{\mathcal{T}}$ are established by a Transformer encoder that follows the seminal paper [27] by using the implementation from the PyTorch [16] library. In practice, we use 4 encoder layers with 6 attention heads and hidden dimensions of size 512. Since the scene is defined as an unordered sequence of its objects and floor representations, we do not additionally encode the position of individual tokens in $\mathcal{T}$, and perform *zero-padding* to handle scenes with different number of objects N. In practice, we consider a maximum number of 50 objects. Following [31], the output token embeddings are each passed to a *shared* MLP that preserves the token dimension d_τ, implemented with a hidden unit of size 768, GELU activation [6], and followed by Layer Normalization [1].

A.1.3. Cross-Attention Layers

As introduced in Section 3.3 of the main paper, the scene conditioning sequence $\hat{\mathcal{T}}$ output by the 3D layout encoder is mapped to associated *key* K^y and *value* V^y by introducing respective learnable dense projection matrices $W_K \in \mathbb{R}^{d \times d_\tau}$ and $W_V \in \mathbb{R}^{d \times d_\tau}$. In practice, we follow the implementation from IP-Adapter [31] to introduce the decoupled cross-attention mechanism within residual blocks of the pretrained Stable Diffusion UNet backbone.

A.2. Datasets

In this section, we introduce the datasets used in our experimental evaluation and describe any dataset-dependent mechanisms implemented to adapt our work to their specific features or available annotations.

A.2.1. HyperSim Dataset

Features Following previous work [30], we leverage indoor scenes and photorealistic renderings from HyperSim [20] to construct a collection of semantic 3D bounding box layouts with camera poses y and image x pairs. The dataset originally features 461 scenes from diverse types of indoor environments, and $77{,}400$ rendered images. We follow SceneCraft [30] and leverage a filtered version of the dataset proposed in [24], discarding unbounded and scenes with excessively large scales. This results in 323 unique scenes, that are associated with $24{,}383$ images annotated by a global text caption extracted by a pretrained foundation model. Each layout features 3D bounding box annotations that are typed according to the standard NYU40 [25] semantic classes. Each rendering is also associated with an object instance map, that we leverage in the context of our SOC metric implementation, as detailed in Section A.6. We conducted foundational statistical analysis of the 3D layouts, establishing that scenes contain an average of 121 objects, with a median of 54 and that the dataset demonstrates a large diversity in terms of structural complexity, which is challenging in the context of our work and in light of the limited number of available samples.

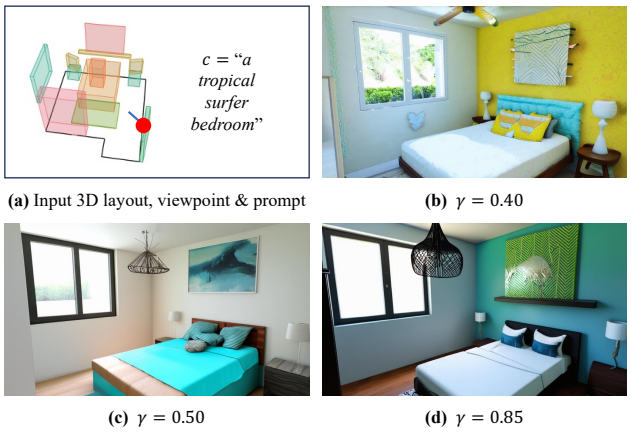
Experimental Setup We describe here the dataset-specific mechanisms implemented to run experiments on HyperSim layouts. Importantly, a proportion of scenes from the dataset contain hundreds of objects, exceeding the maximum length fixed to 50 for our sequence-based representation used for the model conditioning. As a result, we leverage the instance semantic map of the target rendering to prioritize including visible objects in the conditioning sequence. If the number of visible objects still exceeds the limit, we identify the most meaningful objects based on their bounding box dimensions. Since the HyperSim dataset does not provide the floor information in a straightforward manner, we set $\mathcal{F} = \emptyset$ in our experiments. Object-level semantic information are also limited to their categorical label among NYU40 classes, from which we apply $\tau_\theta(s_i)$ as described in our general methodology.

A.2.2. Custom Bedroom Dataset

As described in the main paper in Section 4.1, the limited number of available HyperSim layouts, that are passed as conditioning input, makes it easier for our model to learn a mapping between the global scenes and specific camera poses and target renderings, while overlooking the contributions of individual objects. In response, we gathered a custom dataset of $72{,}000$ human-designed 3D indoor bedroom layouts, in which each layout is associated to a single rendering. The dataset also includes floor information, represented as a 3D point cloud. Finally, ground-truth object-level semantic descriptions s_i are extracted using a LLaVA model [10] from their 2D rendering using the following instruction: *"Describe this object concisely. Do not analyse the background, only the object"*. This level of annotations allows to implement all the building blocks from our general methodology.

A.3. Training Protocol

We optimize our adapter network while keeping the text-to-image denoiser frozen by following standard DDPM [7] training with ϵ-prediction objective, as formalized in Equation (1) in the main paper. Noise timesteps are uniformly sampled from $t \sim \mathcal{U}(0, 1000)$. At each iteration, we randomly drop the conditioning input 3D layout y with a rate $\mathbf{p}_{\text{drop}} = 0.15$. We use the AdamW [11] optimizer, with weight decay coefficient $\lambda = 0.01$ and learning rate $\eta = 5 \times 10^{-5}$, reached following an initial linear *warmup* phase during the first 200 training iterations. Our checkpoints are trained for a total of 250 epochs, with η decreasing according to a cosine schedule. We leverage PyTorch Lightning [4] to distribute experiments across 3 NVIDIA RTX 6000 GPUs, each handling a batch size of 10 samples. Importantly, we experimented with enabling *automatic mixed precision* during training, but empirically observed instability and inconsistent convergence.



(a) Input 3D layout, viewpoint & prompt **(b)** $\gamma = 0.40$ **(c)** $\gamma = 0.50$ **(d)** $\gamma = 0.85$

Figure 2. **Text-driven DiT synthesis results**. Given an input 3D layout, viewpoint and caption (a), LACONIC with Stable Diffusion 3 [3] supports adjusting the adapter strength to balance fidelity to the text prompt against adherence to the input layout (b–d).

A.4. Inference Settings

At test-time, images for quantitative evaluation are generated following the sampling algorithm from DPM-Solver [12]. We apply 30 denoising steps with a classifier-free guidance scale of 5.0. Images in our qualitative results are obtained with the DDIM sampler [26] using 50 steps.

A.5. Baselines

A.5.1. SceneCraft

SceneCraft [30] is a recent baseline method that proposes to tackle 3D layout-guided image synthesis, in which, similar to our approach, 3D layouts are defined as a collection of typed object 3D bounding boxes. To do so, it renders the input 3D layout from the target camera viewpoint to (i) a semantic map in which individual boxes are colored according to their one-hot semantic category and (ii) depth maps that are automatically derived from the input geometry. From this pair of 2D input representations and in order to introduce the additional controls to a pretrained text-to-image backbone, SceneCraft trains dedicated *ControlNet* [32] modules in a supervised experiment. In practice, we leverage the pretrained checkpoint obtained by training the modules on the same HyperSim [20] subset used in our own experiments. It uses Stable Diffusion [21] v2.1 as the T2I prior. We employ official author implementation with the default test-time parameters.

A.5.2. Diffusion Model from Scratch

We describe below the main motivations and implementation details behind the *Diffusion Model trained From Scratch* (DM-FS) from our experimental evaluations.

Intuitions A key design choice in our framework is to use a cross-attention-based *adapter* network to augment existing T2I models with 3D layout guidance, rather than training a dedicated, conditional generative model from scratch. This choice is motivated by two primary factors. First, achieving photorealistic generation results would be extremely challenging given the small scale of available image datasets featuring 3D layout annotations, especially on less common camera views that are not widely represented in the training distribution. Second, our adapter approach allows us to leverage the powerful priors of large-scale T2I models, enabling strong generalization to other domains, as demonstrated by our experimental results. We believe that those insights validate the relevance of this baseline in the context of our contributions' evaluation.

Implementation To experimentally validate our intuitions, we trained a baseline from scratch, jointly optimizing a randomly initialized UNet and the 3D layout encoder from Section A.1.2. The backbone is similar to that of Stable Diffusion, implemented with Diffusers [28], and is guided through standard cross-attention conditioning [21] between the 3D scene sequence $\mathcal{S}$ and feature maps from the residual blocks. Due to the limited caption variety in the training data, we omit text conditioning for this baseline. To account for the smaller training set, the UNet is also downsized compared to Stable Diffusion and comprises 4 downsampling and upsampling residual blocks with 128, 256, 256 and 384 output channels. The two *bottleneck* blocks, that operate at smaller resolutions, are augmented with cross-attention with 8 heads. The 3D layout encoder's architecture is identical to that described in Section A.1.2.

A.6. Evaluation Metric

In Section 4.1 of our main paper, we introduce the Scene Object CLIP (SOC) score as a robust way to simultaneously assess whether objects in the generated content (i) are correctly positioned and sized with respect to their spatial conditioning information and, at the same time, (ii) match their assigned semantic attributes.

Procedure To compute the score for a synthesized image, we first identify its main visible objects given the conditioning 3D layout $\mathcal{S}$, the camera pose $\mathcal{C}$, and the 2D object o_i bounding box annotations $\mathcal{B}^i_{\text{2D}}$ from the associated ground truth image x_0. Since these 2D boxes do not assess that the corresponding objects o_i are mostly visible, we also project the 3D bounding boxes from the conditioning signal onto the image plane to derive enclosing 2D boxes $\mathcal{B}^i_{\text{3D}}$. For each object, we notice that this second bounding box fully contains the one from the image annotation, and compute the ratio r_a as the area of $\mathcal{B}^i_{\text{2D}}$ divided by the area of the enclosing projected box $\mathcal{B}^i_{\text{3D}}$. Remarkably, this projection mecha-

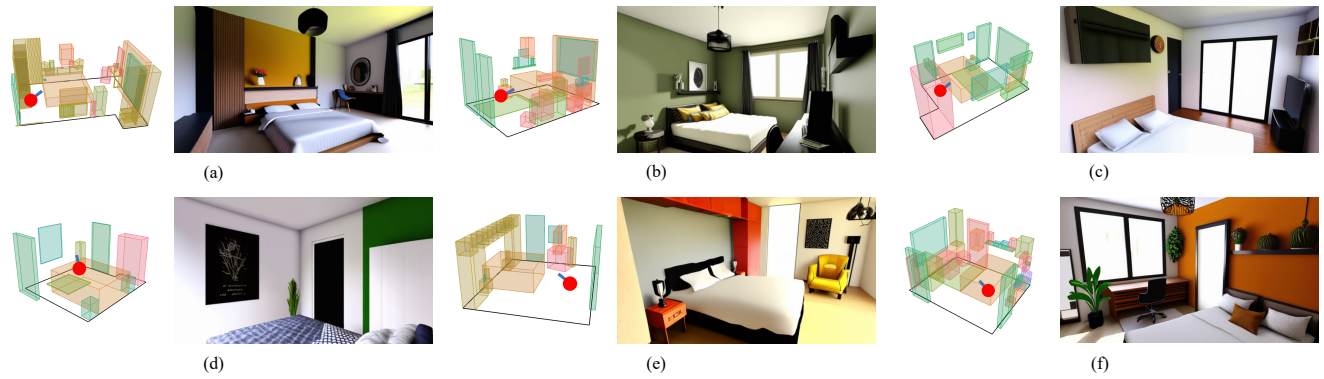


Figure 3. **Layout-guided image synthesis results with a DiT-based backbone**. Our LACONIC adapter successfully conditions Stable Diffusion 3 [3], demonstrating compatibility with modern DiT architectures. The generated images (Right) show strong adherence to the input 3D layouts and camera viewpoints (Left).

nism also handles objects that partially lie outside the image bounds. Based on this ratio, objects o_i that are sufficiently visible are cropped at the location of $\mathcal{B}^i_{\text{3D}}$ in the synthesized image, yielding per-object images x_{o_i}. We finally compute a CLIP [19] correlation score between object images and their associated text semantic s_i.

Filtering Parameters We rely on an object's associated visible area ratio r^i_a to determine if its crop image will be added to the evaluation base. In practice and to report the values in Table 1 in the main paper, we set a threshold value $\alpha = 0.4$. Additionally, we filter out tiny objects, which are prevalent in the HyperSim [20] dataset. Indeed, objects whose instance semantic map covers less than 2% of the image area are discarded. Finally, we also filter objects whose NYU40 semantic annotation is non-descriptive, *e.g.*, that are classified as *"other"*. The final metric is averaged over all the resulting object crops in the dataset.

B. Generalization to DiT Backbones

In this section, we demonstrate the flexibility of our adapter by integrating it with a recent Diffusion Transformer (DiT) [17] backbone. Specifically, we employ Stable Diffusion 3 [3] to showcase that LACONIC is not limited to established UNet-based models conditioned via standard cross-attention, but is capable of remarkable generalization across architectures.

B.1. Background on Joint-Attention

Stable Diffusion 3 [3] introduces a Multimodal Diffusion Transformer (MM-DiT) architecture trained with Rectified Flows [9]. The core of its conditioning approach is a joint-attention mechanism that operates over a unified sequence containing both image tokens (from the latent x_t) and text tokens (from the caption c). Within each transformer block, separate sets of linear projections are used to yield the respective image Q^x, K^x, V^x and text Q^c, K^c, V^c matrices. These are subsequently concatenated into global matrices (*e.g.*, $Q = [Q^x; Q^c]$) used for attention, enabling a bidirectional information flow between text and visual features within the architecture. The resulting hidden states H^x and H^c are finally separated and processed by additional modality-specific feedforward modules.

B.2. Layout Conditioning

Our 3D layout condition is integrated by performing an additional joint-attention pass. Consistent with our general methodology, this is implemented by introducing new trainable linear projections to augment the pretrained model. Specifically, we obtain new key-value pairs from both the text-conditioned image state H^x and the encoded layout y features, while leveraging the query matrix Q from the primary text-conditioning step. A second joint-attention operation is then performed over concatenated matrices, producing an output that is split into a novel image state $H^{x'}$, and a layout state H^y which is processed by additional linear layers. These are merged with the outputs from the primary joint-attention pass, following the linear combination from Equation 3 in the main paper, *i.e.*,

$$H^x_{\text{final}} = H^x + \gamma H^{x'} \quad \text{and} \quad H^c_{\text{final}} = H^c + \gamma H^y \qquad (2)$$

In practice, we leverage the Diffusers backbone implementation and pretrained weights, that we augment with the custom joint-attention processor implementation from IPAdapter-Instruct [22].

B.3. Qualitative Results

We show in Figure 3 images synthesized by Stable Diffusion 3 [3] DiT after training a dedicated LACONIC adapter. Notably, the augmented backbone successfully takes into account the input 3D layout and viewpoint and, as a result,

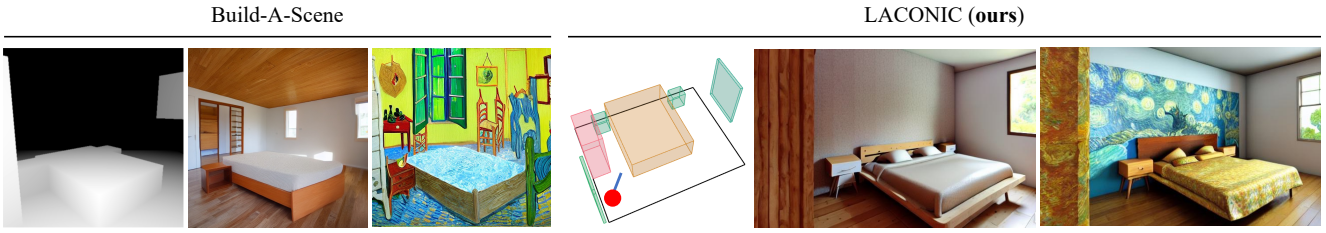


Figure 4. **Qualitative comparison with Build-A-Scene [2] given a common 3D layout and viewpoint**. For each method, from left to right: input scene representation, generation result for prompt c_1 = "*a cozy bedroom with a wooden floor*", and for c_2 = "*a Van-Gogh style bedroom*". Build-A-Scene's input representation also features individual prompt for each bounding box.

renders scenes in accordance with the specified objects spatial and semantic features. Additionally, Figure 2 highlights that the model also supports multimodal conditioning with a non-empty global text caption c, whose influence with respect to the layout can be adjusted following Equation 2.

C. Additional Results

In this section, we provide additional experimental results highlighting the capabilities of our model and advantages of our design choices.

C.1. Ablation Study

We trained ablated versions of our model to measure the individual contribution of:

- **L-R**: applying the reframing mechanism from world to camera coordinates described in Section 3.4.
- **L-T**: including the transformer encoder module in the adapter architecture detailed in Section 3.3.

Quantitative evaluation metrics are computed on the custom *bedroom* dataset and reported in Table 1. Reported results further validate our design choices. Notably, we can observe that the transformation of the input layout to the camera's 3D coordinate system has a key contribution to our method's performance.

Table 1. **Ablation study on our framework's main components**.

Ablation Setting	FID ↓	KID ↓	SOC ↑
L-R	34.03	32.17	22.17
L-T	10.68	11.19	22.28
LACONIC	**9.68**	**10.24**	**22.35**

C.2. Baseline Comparison

We provide an additional qualitative comparison against the recent Build-A-Scene [2] method. While fundamentally different in its setting by adopting a training-free approach, the baseline also attempts to perform 3D layout guided image synthesis. To do so, objects are iteratively added to a given background by leveraging a depth-conditioned image generative model and by manipulating the attention maps of the pretrained backbone. Qualitative results from the same input bedroom layout are reported in Figure 4. Remarkably, our method more accurately reflects the input 3D structure, showcasing all and only the specified objects.

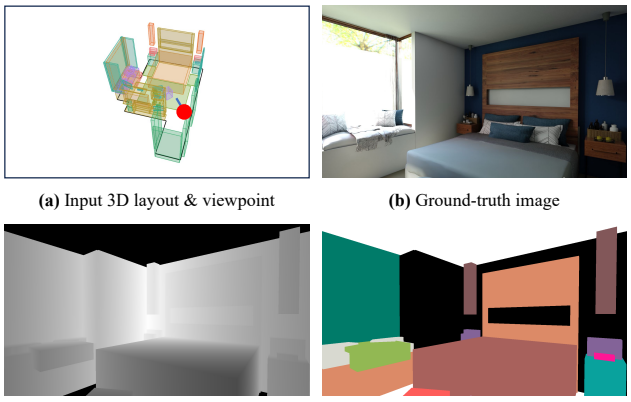
C.3. Perceptual Study

We conducted a perceptual study to compare our method with SceneCraft [30], evaluating both the overall quality of the generated images and their adherence to the input 3D layout and viewpoint.

Baseline Settings Methods are compared on our custom *bedroom* dataset on which we train a SceneCraft model by optimizing jointly respective ControlNet [32] modules from depth and semantic maps. These conditioning inputs are rendered from the 3D layout and at the target camera view, as shown in Figure 5. We follow the official implementation and default parameters, using the same Stable Diffusion v1.5 backbone as our method. A qualitative comparison with this baseline is shown in Figure 8.

Study Design We employed a two-alternative forced-choice (2AFC) test. For a given 3D layout and text prompt, participants were shown the pair of synthesized images and asked to choose the better one based on two separate criteria: (i) "Which image is the most realistic/natural?" and (ii) "Which image better respects the reference 3D layout and viewpoint?". To prevent bias, the positions of the images were randomized for each trial. The study interface provided an interactive view of the 3D layout input and the ground truth image for reference, as shown in Figure 7.

Results We collected 638 preference votes from 15 participants across 319 image pairs. The results demonstrate a clear preference for our method which was favored for realism in 71.2% of comparisons. For adherence to the input 3D layout, LACONIC achieved a strong 89.0% preference rate. This trend was remarkably consistent across participants. Based on their individual average ratings, 100% of participants (15 out of 15) favored our method on conditioning adherence, and 93% (14 out of 15) favored it on realism.



(a) Input 3D layout & viewpoint **(b)** Ground-truth image

(c) Conditioning depth map **(d)** Conditioning segmentation map

Figure 5. **Conditioning inputs for the SceneCraft [30] baseline**. The model is conditioned on depth (c) and segmentation (d) maps, which are rendered from the 3D bounding box layout (a) underlying a ground-truth *bedroom* image (b).

C.4. Qualitative Results

We provide additional qualitative results on text-driven, layout-guided image synthesis in Figure 1. These results highlight our method's ability to adhere to the input semantic 3D layout while achieving out-of-distribution generalization to unseen concepts. Additionally, we report in Figure 9 and Figure 10 supplemental comparisons against the DM-FS and SceneCraft [30] baselines, both with and without providing an additional text condition. Our method consistently generates more realistic images with superior detail, 3D layout fidelity, and text prompt adherence. Motivating our 3D layout conditioning encoder, the DM-FS baseline appears to better respect the input 3D layout in comparison to SceneCraft [30]. We notice that both adapter-based approaches, ours and SceneCraft, leverage the pretrained T2I backbone, resulting in fewer artifacts in the generated images in comparison to DM-FS. Consequently, our method demonstrates a significant advantage in complex scenes with many objects, combining the expressiveness of our 3D layout encoder with the T2I backbone's ability to generate detailed visual features.

D. Limitations

Figure 6 illustrates failure cases and known limitations. Although our method demonstrates state-of-the-art 3D layout adherence, it can occasionally generate results with inconsistencies, such as missing objects, visual artifacts, or distorted perspectives.

E. Societal Impact

Controllable generative AI for 3D environments has significant implications for both individuals and industries. We believe our method to predominantly yield positive societal impact. By enabling fine-grained control over generation, our approach makes 3D-aware content creation more intuitive and accessible, empowering not only artists and designers but also individuals with no prior expertise in 3D modeling. It also benefits industries that rely on realistic environmental rendering, such as urban planning, architecture, and digital twins for cities. A key advantage of our method is its parameter efficiency, which reduces the computational cost of training—an important factor given the high environmental impact of generative models [8, 13]. However, these benefits come with challenges. Since our approach does not natively incorporate recent advances in generative model watermarking [5, 23], it could be misused for potentially deceptive applications. Additionally, by significantly lowering the barrier to content generation, it may contribute to rapid job displacement in traditional 3D design fields. Another concern is the potential for bias propagation, as image generative models, including semantic and style-driven ones like ours, may unintentionally reinforce stereotypes [33]—for example, by associating certain colors with gender, by *e.g.*, synthesizing a bedroom as blue for *"boys"* and pink for *"girls"*. Ensuring the responsible development and use of such technologies, while addressing ethical concerns and mitigating biases, is crucial.

Figure 6. **Failure Cases and Limitations**. In (a), some objects, such as the windows, are missing in the generated image and the floor shape, while complex, is not accurately rendered. We can also observe visual stability issues, as shown in the result in (b), which features an inconsistent pattern on the wall resembling a misplaced frame above the bed. Finally, generated images can exhibit unnatural perspective, resulting in distorted floors and objects, as showcased in (c). This behavior may result from our assumption of consistent camera intrinsics across all data samples.

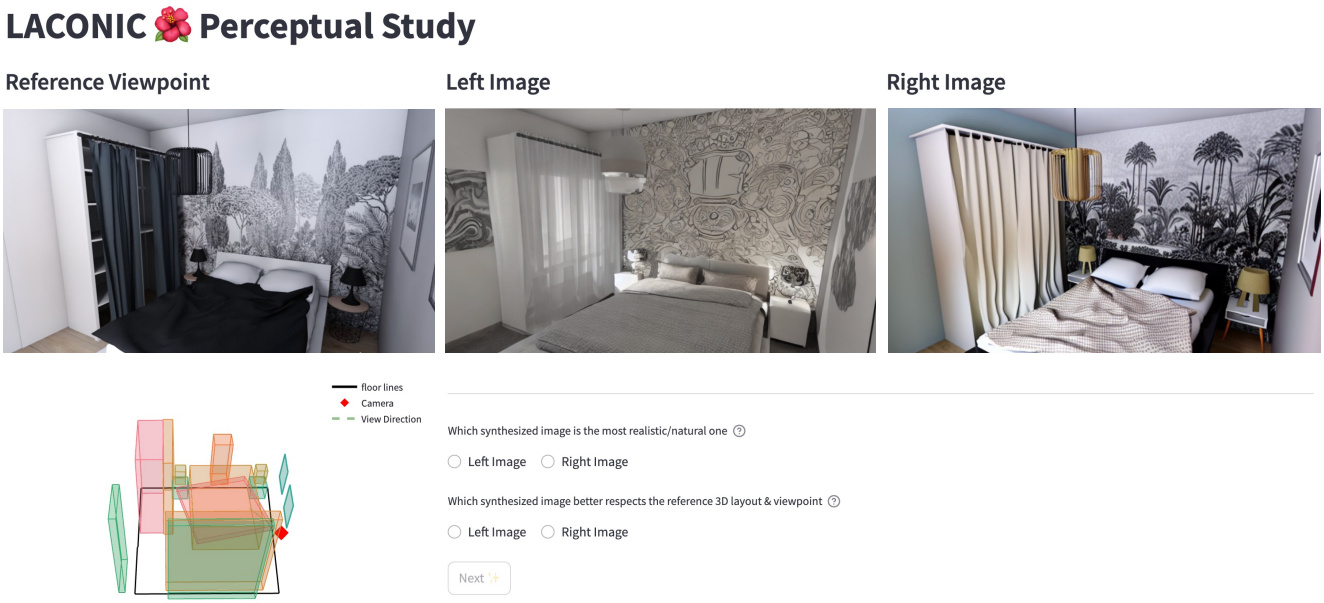


Figure 7. **Perceptual Study Interface**. Users are prompted to independently select which generation result is (i) the most realistic and (ii) more in line with the input 3D layout and viewpoint.

(a) Input 3D Layout **(b)** Ground-Truth Image **(c)** SceneCraft **(c) Ours**

Figure 8. **Comparison with SceneCraft [30] on 3D layout-guided image synthesis.** Our method demonstrates superior realism and adherence to the input 3D layout and viewpoint on our custom *bedroom* dataset.

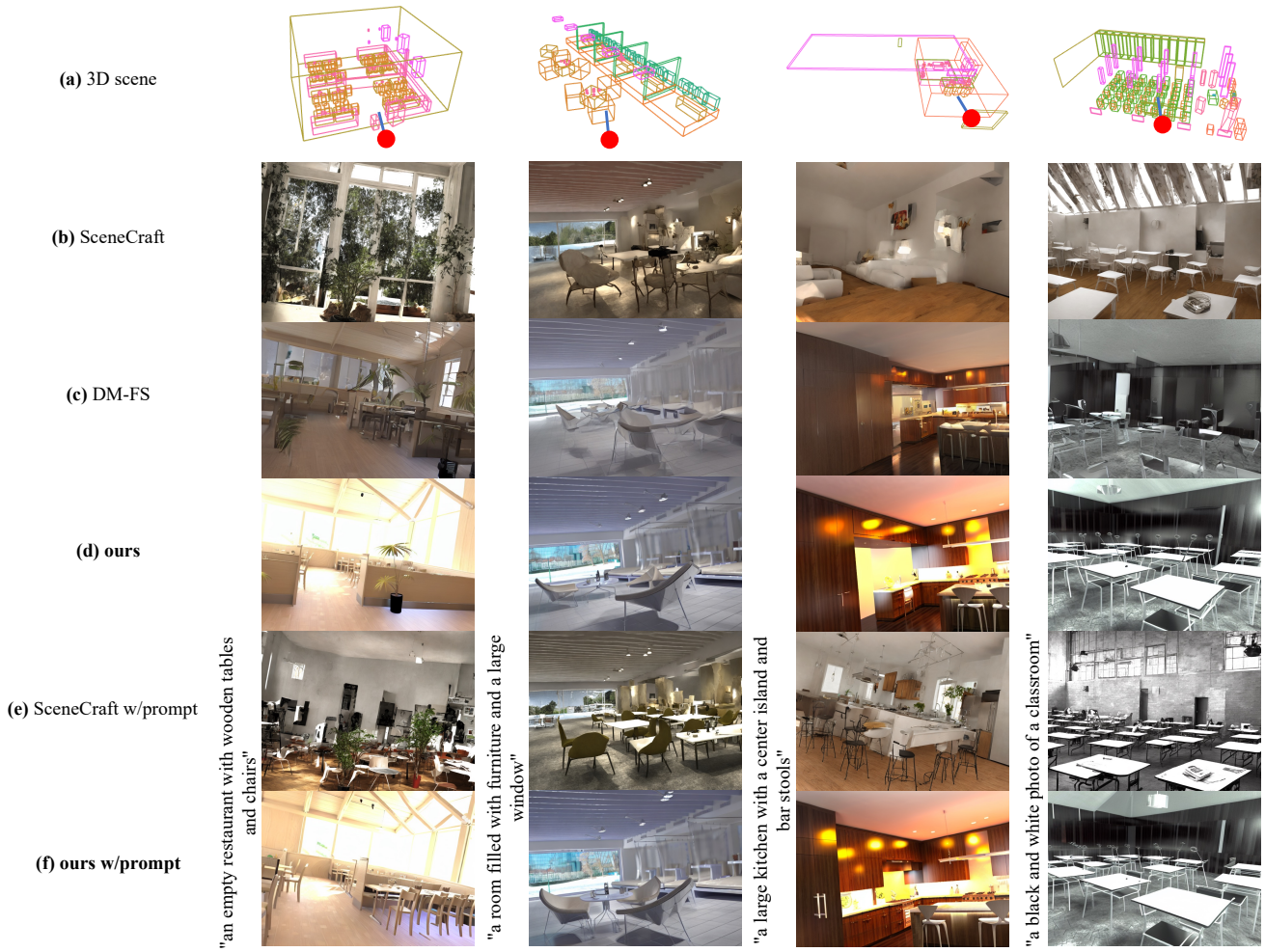


Figure 9. **Additional 3D layout-guided image synthesis baseline comparisons (1/2).** We can observe that our method produces more natural images that better respect the input 3D layout.

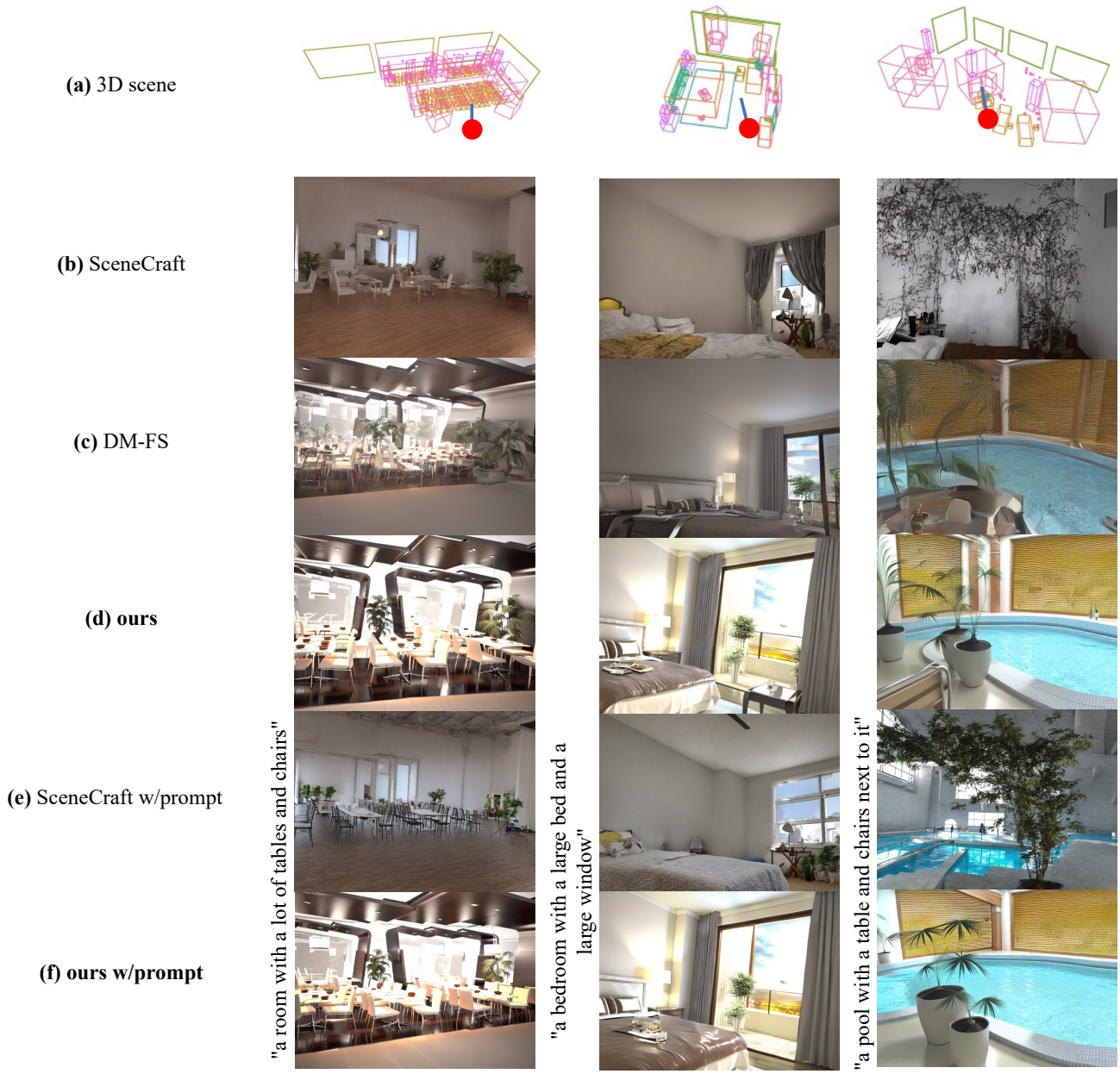


Figure 10. **Additional 3D layout-guided image synthesis baseline comparisons (2/2)**.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2

[2] Abdelrahman Eldesokey and Peter Wonka. Build-a-scene: Interactive 3d layout control for diffusion-based image generation. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. 5

[3] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning (ICML)*, 2024. 3, 4

[4] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 2019. 2

[5] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22466–22477, 2023. 6

[6] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 2

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020. 2

[8] Anne-Laure Ligozat, Julien Lefèvre, Aurélie Bugeau, and Jacques Combaz. Unraveling the hidden environmental impacts of ai solutions for environment. *arXiv preprint arXiv:2110.11822*, 2021. 6

[9] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 4

[10] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:34892–34916, 2023. 2

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[12] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:5775–5787, 2022. 3

[13] Sasha Luccioni, Yacine Jernite, and Emma Strubell. Power hungry processing: Watts driving the cost of ai deployment? In *Proceedings of the 2024 ACM conference on fairness, accountability, and transparency*, pages 85–99, 2024. 6

[14] Léopold Maillard, Nicolas Sereyjol-Garros, Tom Durand, and Maks Ovsjanikov. Debara: Denoising-based 3d room arrangement generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 109202–109232, 2024. 1

[15] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 12013–12026, 2021. 1

[16] 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 (NeurIPS)*, 32, 2019. 2

[17] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023. 4

[18] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017. 1

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 1, 4

[20] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10912–10922, 2021. 2, 3, 4

[21] 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 (CVPR)*, pages 10684–10695, 2022. 1, 3

[22] Ciara Rowles, Shimon Vainer, Dante De Nigris, Slava Elizarov, Konstantin Kutsy, and Simon Donné. Ipadapter-instruct: Resolving ambiguity in image-based conditioning using instruct prompts. In *European Conference on Computer Vision (ECCV)*, pages 54–70. Springer, 2025. 4

[23] Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. *arXiv preprint arXiv:2411.07231*, 2024. 6

[24] Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, et al. Controlroom3d: Room generation using semantic proxy rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6201–6210, 2024. 2

[25] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 2

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 1, 2

[28] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 1, 3

[29] Qiuhong Anna Wei, Sijie Ding, Jeong Joon Park, Rahul Sajnani, Adrien Poulenard, Srinath Sridhar, and Leonidas Guibas. Lego-net: Learning regular rearrangements of objects in rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19037–19047, 2023. 1

[30] Xiuyu Yang, Yunze Man, Junkun Chen, and Yu-Xiong Wang. Scenecraft: Layout-guided 3d scene generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:82060–82084, 2024. 2, 3, 5, 6, 7

[31] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 2

[32] 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 (ICCV)*, pages 3836–3847, 2023. 3, 5

[33] Mi Zhou, Vibhanshu Abhishek, Timothy Derdenger, Jaymo Kim, and Kannan Srinivasan. Bias in generative ai. *arXiv preprint arXiv:2403.02726*, 2024. 6

[34] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5745–5753, 2019. 1