Collaborative Control for Geometry-Conditioned PBR Image Generation

Shimon Vainer[1,⋆], Mark Boss[2,†], Mathias Parger[1], Konstantin Kutsy[1], Dante De Nigris[1], Ciara Rowles[1], Nicolas Perony[1], and Simon Donné[1,⋆]

Unity Technologies[1] Stability AI[2]
⋆ Equal Contributions † Core Technical Contributions
Corresponding author: shimon.vainer@unity3d.com

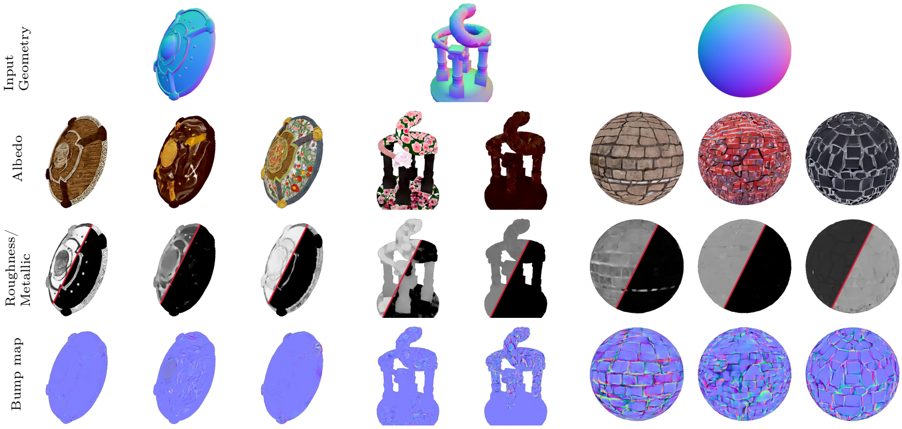


Fig. 1: Generated PBR materials. By tightly linking the PBR diffusion model with a frozen RGB model, we produce high-quality PBR images conditioned on geometry and prompts. Visit the project page at https://unity-research.github.io/holo-gen.

Abstract. Current 3D content generation builds on generative models that output RGB images. Modern graphics pipelines, however, require physically-based rendering (PBR) material properties. We propose to model the PBR image distribution directly to avoid photometric inaccuracies in RGB generation and the inherent ambiguity in extracting PBR from RGB. Existing paradigms for cross-modal finetuning are not suited for PBR generation due to both a lack of data and the high dimensionality of the output modalities: we overcome both challenges by retaining a frozen RGB model and tightly linking a newly trained PBR model using a novel cross-network communication paradigm. As the base RGB model is fully frozen, the proposed method does not risk catastrophic forgetting during finetuning and remains compatible with techniques such as IPAdapter [73] pretrained for the base RGB model. We validate our design choices, robustness to data sparsity, and compare against existing paradigms with an extensive experimental section.

Keywords: Image Generation, Material Properties, Multi-Modal Generation

1 Introduction

The recent meteoric rise of diffusion models has made automated at-scale generation of high-quality RGB image content more accessible than ever. Continuing on this success, Text-to-Texture and Text-to-3D approaches successfully lifted the generation to 3D [33]. Yet to maximize the usefulness of the generated textures in downstream 3D workflows, they must be compatible with physically-based rendering (PBR) pipelines for proper shading and relighting. Current approaches to PBR texture generation rely on generated RGB images and subsequent PBR extraction through an inverse rendering process, facing physically **in**accurate lighting in the generated RGB diffusion images as well as significant ambiguities in the inverse rendering. We propose a solution for geometry-conditioned generation of PBR images by modeling the joint distribution directly, avoiding the issues around photometric consistency and inverse rendering.

To model the distribution of non-RGB modalities, existing approaches typically fine-tune the weights of a base RGB model. Applied to PBR images, this implies either directly predicting the entire PBR image stack or sequentially predicting them conditioned on one another. Neither is sufficient for our use-case: jointly predicting the entire PBR image stack is problematic as the higher-dimensional modality does not compress well into the established latent spaces, and sequentially predicting the elements of the PBR image stack is significantly more expensive and risks compounding errors in the sequential generation. Furthermore, while state-of-the-art RGB diffusion models are trained on billions of images [51], there is unfortunately no dataset of such size at our disposal for PBR content generation. Instead, the largest available dataset of PBR content is Objaverse [10], containing around 800,000 objects with associated PBR textures. In light of the restricted training data available, finetuning the base model results in catastrophic forgetting, forfeiting generalizability, as we illustrate in Sec. 5.

Instead, we keep the pre-trained RGB image model frozen and train a parallel model to generate the PBR image stack, as shown in Fig. 2. Using our proposed cross-network control paradigm, we tightly link the PBR model to the frozen RGB model in order to leverage its expressivity and rich internal state. As a result we are able to generate qualitative and diverse PBR content, even far out-of-distribution for the Objaverse dataset. Crucially, the frozen RGB model safeguards against catastrophic forgetting *and* remains compatible in a plug-and-play fashion with techniques such as IPAdapter [73]. In summary, we:

1. Propose the novel *Collaborative Control*paradigm to tightly link the PBR generator to a fully frozen pre-trained RGB model, modeling the joint distribution of RGB and PBR images directly (see Sec. 4.1),
2. Illustrate that the proposed control mechanism is data-efficient, and generates high-quality images even from a very restricted training set,
3. Ablate our design choices to show the improvement over existing paradigms in literature and the issues with existing approachs, and
4. Demonstrate the compatibility with IPAdapter [73] specifically.

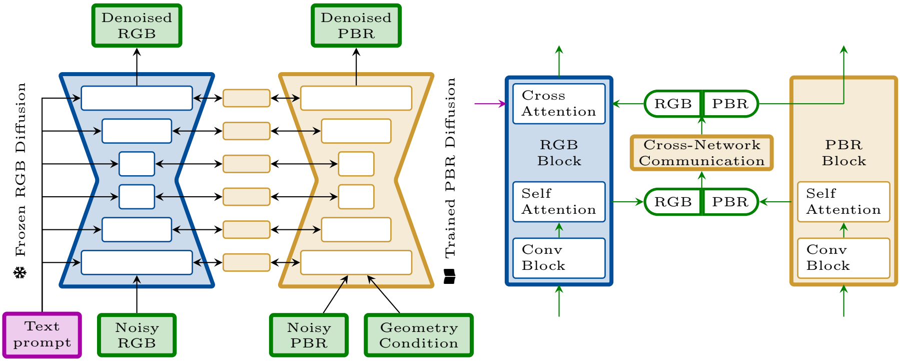


Fig. 2: Collaborative Control. Two parallel models collaborate to generate pixel-aligned outputs of different modalities. We freeze the left pre-trained RGB model and train the right PBR model with its cross-network communication layers. Communication is achieved by concatenating the states of both models, processing them with a single linear layer for a residual update, and distributing the result back to the respective models. As discussed in Sec. 5, prompt cross-attention in the PBR model is counter-productive.

2 Related Work

Generating natural images from text prompts Natural image generation has a long history: from GANs [14,24–27] and VAEs [29], to auto-regressive models [44,63] and more recently diffusion models [18,56]. Although conditioning GAN networks on text prompts or other embeddings has proven difficult [47], recent work continues to improve GAN-based text-to-image generation [50]. The introduction of diffusion models [18,56], which model the generation process as the iterative inverse of an degeneration process [55], was a breakthrough in the generative field — it is far more stable than typical GAN training, albeit slow and computationally expensive.

Unfortunately, these approaches require internet-scale data to train from scratch [51]. For PBR image generation, the largest commonly available dataset is Objaverse [10]: at 800,000+ objects it is still several orders of magnitude smaller than LAION-5B [51] and proves insufficient to train high-quality generative models from scratch (as illustrated by the finetuning baseline failing to generalize in Sec. 5). While pre-trained RGB models encode rich prior knowledge around structure, semantics, and materials [52,58], Sarkar et al. warn that the models are often still inaccurate when it comes to structure [49]. We argue that this extends to material properties: diffusion models prefer idealized and artistic appearances, rather than physically accurate scenes. Therefore, we now discuss how to extract relevant knowledge out of the powerful pre-trained RGB models.

Leveraging pre-trained RGB models to generate other modalities The authors of Marigold [28] interpret depth maps as grayscale images to leverage a pre-trained LDM and its linked VAE, while the authors of DiffusionDepth [12] train a VAE and diffusion model from scratch. Alternatively, Zhao et al. [80] extract multi-scale feature maps from a pre-trained diffusion model's activations to decode for a variety of tasks, such as monocular depth prediction. For depth prediction, LDM3D [57] shows that it is possible to fine-tune the VAE to compress the joint RGBD image into the original latent space's dimension, which roughly preserves the latent space so that finetuning the base diffusion model for joint RGBD diffusion is tractable. Neither approach is plausible for PBR image generation: compressing the PBR data together with RGB in the original latent space overloads the capacity of that low-dimensional latent space, and splitting the PBR image stack up into separate triplets that the diffusion model can directly predict is costly as they need to be generated sequentially to properly model the joint distribution. Wonder3D [39] and UniDream [38] perform joint RGB and normal diffusion with parallel branches using a cross-domain self-attention that aligns both branches; yet the number of diffusion models scales linearly, and the cross-domain self-attention quadratically, with increasing number of output modalities. Our proposed approach differs in two key aspects from these multi-modal paradigms: (1) we do not fine-tune the base RGB model at all, reducing the risk of catastrophic forgetting as much as possible, and (2) we train a joint parallel branch for all additional modalities, reducing cost. For the latent encoding, we train a PBR-specific VAE to encode all PBR channels jointly.

Conditioning on images Image-based conditioning of diffusion models is relevant to us for two reasons: the similarity between conditioning paradigms and our proposed Collaborative Control, and the fact that we tackle geometry-conditioned PBR image generation. Existing pixel-accurate control techniques come in two flavours: re-training of the base model with modified input and output spaces [12,28], and training of a parallel model that affects the base model's state [11,19,79]; we have found that the former risks catastrophic forgetting of the base model's expressiveness and quality. As in the latter techniques, we leave the base model's weights fully frozen and residually consume and edit the internal states from a parallel model. In ControlNet-XS [11], the authors investigate the connectivity between the base model and the control model, concluding that a connection from the base model's encoder to the controlling model's encoder, and from there to the base decoder, provides sufficient information flow for optimal performance. Yet in both ControlNet and ControlNet-XS, the controlling model only influences the base RGB model's output. In AnimateAnyone [19], the parallel model hooks into the state of the base RGB model to output its own new RGB image instead. For the purpose of our joint RGB-PBR diffusion model, our PBR branch both *controls* the base model (to keep it aligned during the iterative generation process), and *generates* its own PBR output based on the RGB model's internal state; therefore, our proposed approach requires two parallel branches with full bi-directional connections between them. Please refer to Fig. 5 for a schematic overview of these methods.

To condition on input geometry we adopt the choice by Ke et al. [28] of concatenating the conditioning to the PBR model's inputs. As we are training the PBR model from scratch in any case, this does not introduce additional cost.

Text-to-3D describes the task of generating full 3D models from text prompts, often with the aim to support downstream graphics pipelines such as game engines. Earlier methods leverage a pre-trained RGB diffusion model to extract direct appearance, typically leveraging Score Distillation Sampling [45] (SDS) to iteratively optimize a 3D representation by backpropagating the diffusion model's noise predictions [8,15,21,35,40,41,54,60,61,66–68,71,81,82], or building on viewpoint-aware image models [20,36,37,39,46,53,76] to perform direct fusion. Other authors have investigated distilling a pre-trained image RGB model into a 3D generative technique [42,65]. These RGB methods ignore view-dependency of objects, often resulting in artefacts around highlights, and do not result in a representation that is useful in graphics pipelines. Work that does generate PBR properties does so by backpropagating the denoised predictions through a differentiable renderer [7,38,70,72,74]. For such methods, and inverse rendering in general, a major concern is lighting being baked into the material channels: HyperDreamer [70] introduces an ad-hoc segmentation [30]-based regularization loss on the albedo channel to reduce artefacts. By directly generating PBR content rather than going through an inverse rendering process, our proposed technique could resolve many of the issues related to the inverse rendering in the latter methods.

Text-to-Texture methods restrict the Text-to-3D problem to objects with known structure, typically by conditioning the diffusion model on the object geometry [4,6,31,32,75,75,77,77,78]. Paint3D [77] also discusses the lighting artefacts typical with inverse rendering and introduces a custom post-processing diffusion model to alleviate these. Here, too, our proposed approach directly models and generates the full PBR image distribution, and could help these technique side-step the issues with inverse rendering completely.

Evaluation metrics for generative methods When evaluating generative methods, we wish to compare the output distributions with known ground-truth distributions. This is typically done by use of the Inception Score [48] and the Fréchet Inception Distance [17], which compare hidden state distributions of Inceptionv3 [59] between both image sets, as directly modeling the distributions of the high-dimensional images is not tractable. The authors of CMMD [22] argue that neither of these metrics is well suited to modern generative models, and propose a new metric that compares the distributions of CLIP embeddings of the generated images, showing that it aligns well with human observers, especially in respect to low-level image degradations.

Aside from comparing the modelled distributions with the ground truth distributions, we also wish to evaluate the general quality of the generated images for text prompts that are out of distribution for the training dataset. The CLIP score [16] compares the CLIP image embedding with the embedding of the prompt, indicating how well the text prompt was followed: whether all the relevant elements are represented and whether no extraneous elements were introduced. We also report the OneAlign *aesthetics* and *quality* metrics of the generated images [69], which have been shown to align well with human perception, to provide a more quantitative indication of quality.

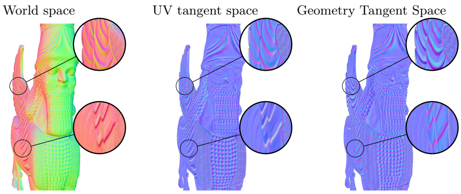


Fig. 3: Bump map. Similar surface bumps in world space (left) are dissimilar in the UV tangent space (middle) because of the arbitrary UV mapping. Representing the bump map in a tangent space solely dependent on the geometry (right) resolves this issue.

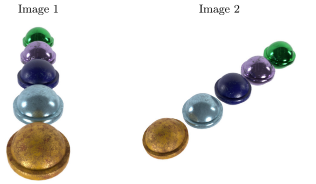


Fig. 4: Rendering function. The dataset is constructed so that the lighting remains constant with respect to the camera. This simplifies the rendering function f_{RGB}. Notice the similar highlight location.

3 Preliminaries

PBR materials are a compact representation of the bi-directional reflectance distribution function (BRDF), which describes how light is reflected from the surface of an object. We use the popular Cook-Torrance analytical Bi-directional Reflectance Distribution Function (BRDF) model [9], more specifically, the Disney BRDF Basecolor-Metallic parametrization [3] as it inherently promotes physical correctness. In this parametrization, the BRDF comprises *Albedo* ($\boldsymbol{b}_a \in \mathbb{R}^3$), *Metallic* ($\boldsymbol{b}_m \in \mathbb{R}$), and *Roughness* ($\boldsymbol{b}_r \in \mathbb{R}$) components. To increase realism during rendering beyond the resolution of the underlying geometry, graphics pipelines add small details such as wood grain or grout between tiles by encoding them in a bump map $\boldsymbol{b}_n \in \mathbb{R}^3$. As this bump map is typically defined in a tangent space based on an arbitrary UV-unwrapping, it entangles the surface property with this arbitrary UV mapping. Instead, we propose to predict the bump map defined in a tangent space based solely on the object geometry, disentangling the texture from the UV mapping as shown in Fig. 3. To construct this geometry tangent space for a point $\boldsymbol{p} = [p_x, p_y, p_z]^T$ with geometry normal $\boldsymbol{n}$, we construct the local tangent vector as $\boldsymbol{t} = \boldsymbol{n} \times ([-p_y, p_x, 0]^T \times \boldsymbol{n})$, corresponding to Blender's *Radial Z* geometry tangent. The geometry tangent space is then constructed as $(\boldsymbol{t}/\|\boldsymbol{t}\|, \boldsymbol{n} \times \boldsymbol{t}/\|\boldsymbol{t}\|, \boldsymbol{n})^T$.

Diffusion models [18,55] iteratively invert a forward degradation process to generate high-quality images from pure noise (typically white gaussian noise). Formally, the forward process iteratively degrades images from the data distribution $\boldsymbol{z}_0 \sim p(\boldsymbol{z})$ to standard-normal samples $\boldsymbol{z}_T \sim \mathcal{N}(0, I)$ over the course of T degradation steps as $\boldsymbol{z}_t \sim \mathcal{N}(\alpha_t \boldsymbol{z}_{t-1}, (1-\alpha_t)I)$, where α_t denotes the noise schedule for timestep t. Practically, the forward process can be condensed into the direct distribution $\boldsymbol{z}_t \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{z}_0, (1-\bar{\alpha}_t)I)$ with the appropriate choice of $\bar{\alpha}_t$. The diffusion model $\mathcal{D}$ is trained to sample the stochastic reverse process $\mathcal{D}_t(\boldsymbol{z}_t) \sim p(\boldsymbol{z}_{t-1}|\boldsymbol{z}_t)$.

4 Method

We wish to train a PBR diffusion model $\mathcal{D}_{pbr}$ that models the reverse denoising process for PBR images as represented in the latent space of a VAE [47]. It represents the data distribution $p(\boldsymbol{z}_{pbr})$. We find that we lack the data required to train this model directly, and instead propose to model $p(\boldsymbol{z}'_{rgb} := f_{rgb}(\boldsymbol{z}_{pbr}), \boldsymbol{z}_{pbr})$ based on an RGB diffusion model $\mathcal{D}_{rgb}$ for the RGB data distribution $p(\boldsymbol{z}_{rgb})$; f_{rgb} is a rendering function that projects the PBR images onto the RGB domain. To motivate this, we split the joint reverse process into two separate processes:

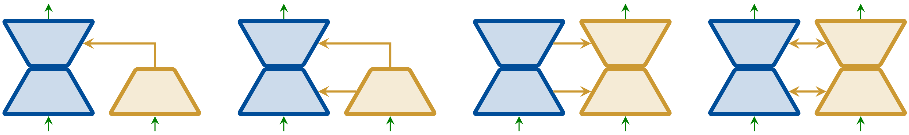
$$\begin{aligned} &p(\boldsymbol{z}'_{rgb,t-1}, \boldsymbol{z}_{pbr,t-1} | \boldsymbol{z}'_{rgb,t}, \boldsymbol{z}_{pbr,t}) \\ &\quad \sim p(\boldsymbol{z}'_{rgb,t-1} | \boldsymbol{z}'_{rgb,t}, \boldsymbol{z}_{pbr,t}) p(\boldsymbol{z}_{pbr,t-1} | \boldsymbol{z}'_{rgb,t-1}, \boldsymbol{z}'_{rgb,t}, \boldsymbol{z}_{pbr,t}) \end{aligned} \tag{1}$$

The RGB model is implemented based on $\mathcal{D}_{rgb}(\boldsymbol{z}_{rgb,t-1}) \sim p(\boldsymbol{z}_{rgb,t-1} | \boldsymbol{z}_{rgb,t})$: by adjusting the internal hidden states based on the PBR model's state, we align the current RGB sample with the PBR sample and restrict it to $\text{Im}(f)$[1]. To simplify this alignment problem, the rendering function f_{rgb} uses fixed camera settings and a fixed environment map as shown in Fig. 4. The PBR model now no longer models $p(\boldsymbol{z}_{pbr,t-1} | \boldsymbol{z}_{pbr,t})$: it additionally has access to the RGB context $(\boldsymbol{z}'_{rgb,t-1}, \boldsymbol{z}'_{rgb,t})$ which simplifies the problem. The RGB and PBR models are in practice much more intertwined than Eq. (1) implies: this derivation serves mostly as an intuitive indication for why the joint problem is more tractable. Note that $\boldsymbol{z}'_{rgb,t}$ is a degraded version of $\boldsymbol{z}'_{rgb,0}$, and *not* a rendered version of $\boldsymbol{z}_{pbr,t}$: the PBR model does not learn to do inverse rendering in degraded image space but rather learns to denoise PBR images given additional RGB context.

4.1 Collaborative Control

In summary, our proposed approach comprises two models working in tandem: a pre-trained RGB image model and a new PBR model, tightly linked to one another (see Fig. 5 for a high-level overview of our proposed control scheme). The previous section identifies two tasks for this cross-network communication: aligning the RGB model's output with both the PBR model's output and the map of the rendering function $\text{Im}(f)$, and communicating knowledge in the RGB model to the PBR model. ControlNet [79] and ControlNet-XS [11] discuss solutions to the former control problem — the authors conclude that communication from the base model's encoder to the controlling model's encoder, and from the controlling model's decoder to the base model's decoder, is sufficient. AnimateAnyone [19] addresses the latter problem and concludes that, there, uni-directional communication from the left model to the right model is sufficient. We have found that for our problem, full bi-directional communication between both networks is crucial; we dub this *Collaborative Control*. See Fig. 5 for a visualization of these control schemes.

[1] Early in training, generated sample quality can be boosted significantly by applying the foreground mask to the RGB estimate for the first few timesteps; a rough projection to bring the estimate much closer to $\text{Im}(f)$. After longer training, this is no longer necessary.



(a) ControlNet (b) ControlNet-Xs (c) AnimateAnyone (d) Proposed

Fig. 5: High-level overview of communication paradigms in (a) ControlNet [79], (b) ControlNet-XS [11], (c) AnimateAnyone [19] and (d) our proposed Collaborative Control approach. Blue represents frozen blocks, while orange elements are optimized during training.

We implement the cross-network communication as a connecting layer between the two models after every self-attention module; its inputs are the concatenation of the model states and its outputs are residually distributed to both models again. During training, we only optimize the weights of the PBR model and the cross-network communication links against both models' outputs, while the RGB model remains fully frozen. By adopting this approach, we safeguard the base RGB model's weights, and do not risk catastrophic forgetting for that base model. As we discuss in Sec. 5, we have found that a single per-pixel linear layer is sufficient, although we also evaluate the other control schemes from Fig. 5 as well as an attention-based communication layer. Notably, we have also found that disabling the text cross-attention in the PBR model is crucial to out-of-distribution performance; we attribute this to overfitting on the restricted dataset, as this problem worsens with reduced training data.

4.2 Implementational details

PBR images latent encoding RGB diffusion models benefit immensely from a dedicated VAE to downsample the images into a lower-dimensional latent space [47]. Existing solutions that generate an alternate modality typically encode that modality with the RGB VAE, but PBR images cannot be compressed into the same latent space due to the higher dimensionality. Instead, we could select channel triplets $\boldsymbol{b}_a$, $[\boldsymbol{b}_m, \boldsymbol{b}_r, \mathbf{0}]$, and $\boldsymbol{b}_n$ and process those with the RGB VAE, but we instead choose to train a dedicated PBR VAE — our ablation studies indicate that the distribution mismatch between the PBR channels and the RGB space is too large, and performance suffers. We adopt the VAE architecture and training code from StableDiffusion v1.5 [47], although, following Vecchio et al. [64], we set the latent space channel count to 14 to maintain optimal balance between quality and compression when processing PBR images.

Conditioning on existing geometry As we wish to generate PBR images conditioned on existing objects, we concatenate the screenspace geometry normals to the PBR model's inputs. Referring to Fig. 5, we note that Collaborative Control encapsulates the ControlNet scheme that would typically be used for this conditioning [11]. As we jointly train both the Collaborative Control scheme and the geometry conditioning from scratch, this does not introduce additional cost.

Generating training data Our dataset for training both the PBR VAE and the Collaborative Control scheme is based on Objaverse [10]: a dataset containing 800,000+ 3D models with annotations for what the models represent (describing both shape and texture). After sanitizing and filtering the dataset we retain roughly 300,000 objects. Each of the objects is rendered with Blender 2.35 from 16 viewpoints encircling the object using a fixed pinhole camera model and a fixed (camera colocated) environment map[2] as in Fig. 4. For the evaluations in Sec. 5, we randomly select 2% of the generated images for evaluation purposes.

Training Collaborative Control For most of the experiments in Sec. 5, ZeroDiffusion [34,62] is the base RGB model, a zero-terminal-SNR version fine-tuned from StableDiffusion v1.5 [47]. As Collaborative Control is agnostic to the base model, we also illustrate StableDiffusion v1.5 and v2.1 as base models in Sec. 5. We optimize the PBR model's weights as well as the cross-network communication layers to minimize the training loss for the RGB and PBR denoising jointly, while keeping the RGB model fully frozen. In almost all cases, we train directly on the final output resolution of 512×512 for a total of 200 000 update steps with a batch size of 12 with a learning rate to $3e^{-5}$ (run on a single A100, taking roughly two days per modal). We also evaluate the effect of a larger training budget by training the proposed model also on 8A100s for the same amount of steps, increasing the batch size by a factor of 8 without affecting training time — for environmental and cost purposes, the training budget is kept low for the main ablation study.

5 Results

Distribution match metrics As an evaluation of how well the data distribution is modeled, distribution match is considered a proxy to both quality and diversity. The Inception Score (IS [48]), which checks the distribution match against ImageNet, is not relevant in a PBR context as it applies only to RGB images. The Fréchet Inception Distance (FID [17]), which compares the distributions of the last hidden state of the Inceptionv3 [59] network on both real and generated images, has been found to better align to perceptual quality. Finally, the recently introduced CLIP Maximum-Mean Discrepancy (CMMD [22]) compares the distributions of the CLIP embeddings of both generated images and a reference dataset. It offers significantly improved sample efficiency, and was shown by the authors to be a better indicator of low-level image quality than FID. However, these metrics are intended for three-channel color images; to use them to model higher-dimensional distributions, we use the technique by Chambon et al. [5] to average the relevant scores of multiple triplets. Here we report as PBR distribution match the average of these scores over each of the channels independently as well as the (grayscale albedo, roughness, metallic), (roughness, metallic, normal XY norm) and (grayscale albedo, normal X, normal Y) triplets, as the full set of triplets is prohibitively expensive to compute. Please refer to the supplementary material for all of the constituting scores.

[2] `https://polyhaven.com/a/studio_small_08`

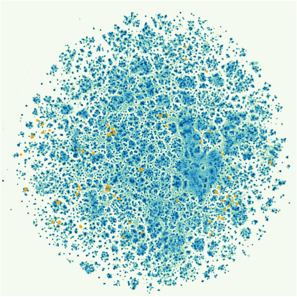


Fig. 6: tSNE visualization of CLIP embeddings of all prompts in Objaverse (blue) and our OOD prompts (orange).

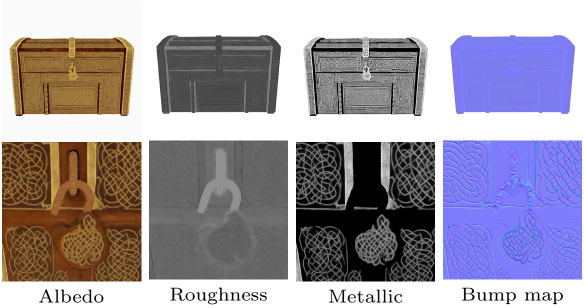


Fig. 7: Despite training on 512 × 512, this PBR model is able to produce full quality results in the trained resolution 768 × 768 of its base StableDiffusion 2.1 RGB model. If that does not suffice, our model also performs well on zoomed-in regions.

Out-of-distribution (OOD) performance metrics indicate the level to which our generator can align to conditioning which it was not trained on. Recent work has introduced the CLIP alignment score [13,16], which estimates the average distance between the text prompt CLIP embedding and the generated image's CLIP embedding, indicating how faithfully the prompt was followed. Additionally, OneAlign [69] is a neural model that estimates aesthetics and quality of input images, shown to align well with human opinions. For the OOD comparison, we hand-pick a subset of 50 every-day objects from Objaverse and ask ChatGPT4 [1] to provide 5 unlikely appearances for each object. Figure 6 illustrates where these OOD prompts lie in the CLIP embedding space using a tSNE visualization.

5.1 Comparisons and Ablations

To the best of our knowledge, there are no PBR generation models in literature that generate PBR for entire scenes (only for generation of single materials [64]). Therefore, we perform an extensive ablation study on our design choices, taking care to include typical approaches from techniques that generate other modalities than PBR. Please refer to Fig. 8 for a qualitative comparison between the model variants, while Tab. 1 contains the quantitative results.

Comparison between control paradigms We compare the performance of the proposed bi-directional cross-network communication layer against two other paradigms: one inspired by AnimateAnyone [19], and one inspired by ControlNet-XS [11]. In the former, dubbed *one-way* communication, the communication layers receive as input only the RGB model's internal state, and they only affect the PBR model's internal state. The latter, dubbed *clockwise* communication, functions in the same way for the encoder part of the architecture, but reverses the information flow to instead only allow editing of the RGB model's internal state based on the PBR model's state. We see that the *one-way* attention does not perform well, with lower distribution match scores as well as OOD performance scores; we attribute this to the fact that the frozen RGB model cannot align to the conditional distribution required from it in Eq. (1) without controlling its internal state. The *clockwise* attention performs significantly better, but is likely still hampered by $\boldsymbol{z}'_{rgb,t-1}$ not being easily available to the PBR model — a similar reasoning as to why the authors of ControlNet-XS included the direct communication link between the base and controlling models' encoders.

Comparison against finetuning We also compare Collaborative Control against the alternative where we edit the first and last layers of the pre-trained network to match the dimensionality of the PBR images (optionally with the rendered image), and then fine-tune the entire network end-to-end. Although the distribution match scores for these finetuning variants are similar to Collaborative Control, the finetuning methods strongly overfit to the training data and perform very poorly in a qualitative comparison.

Comparison between communication types In terms of the type of communication, we compare the proposed single-layer per-pixel communication against a per-pixel MLP-based communication layer, and a global attention layer. The latter performs surprisingly well considering that it disregards pixel locations completely; it is hard to enforce pixel-wise alignment through a global attention layer, which we hypothesize to be the reason for the lower quantitative performance. As Jin et al. [23] discuss, an attention-based architecture is also less robust to resolution changes. The per-pixel MLP, with four hidden per-pixel linear layers with normalization layers [2] in-between, does not qualitatively perform notably better than the single-layer communication layer, so that we settle for the simpler and more computationally efficient choice.

PBR-specific VAE vs RGB VAE We compare the performance of Collaborative Control with a PBR-specific VAE against a version that uses the triplets-based RGB VAE from Sec. 4 to encode the PBR channels (encoding albedo, roughness+metallic, and bump maps in separate triplets and concatenating their latent representations). The mismatch between the PBR domain is clear, both quantitatively in the worse distribution matching scores, and qualitatively in the produced images.

Impact of the training budget Comparing the version training on a single A100 with the version trained with 8 A100s (for eight times the batch size), we see that the latter performs significantly better quantitatively in terms of distribution match, but not quality. Visually, the differences are less clear, although the higher budget model appears to follow complex prompts slightly better.

Impact of the training resolution We compare the performance of Collaborative Control with two training resolutions: 256×256 and 512×512, with an evaluation resolution of 512×512 (which is also what the ZeroDiffusion base model was trained on). While quantitatively, the low-resolution model appears to perform better, visually it is clear that it does not capture the same level of detail as the high-resolution model — we blame this on the metrics not capturing low-level image quality well, focusing instead on high-level encoding of the images. Not that deviating significantly from the training resolution of the base model leads to degraded performance in all cases [23].

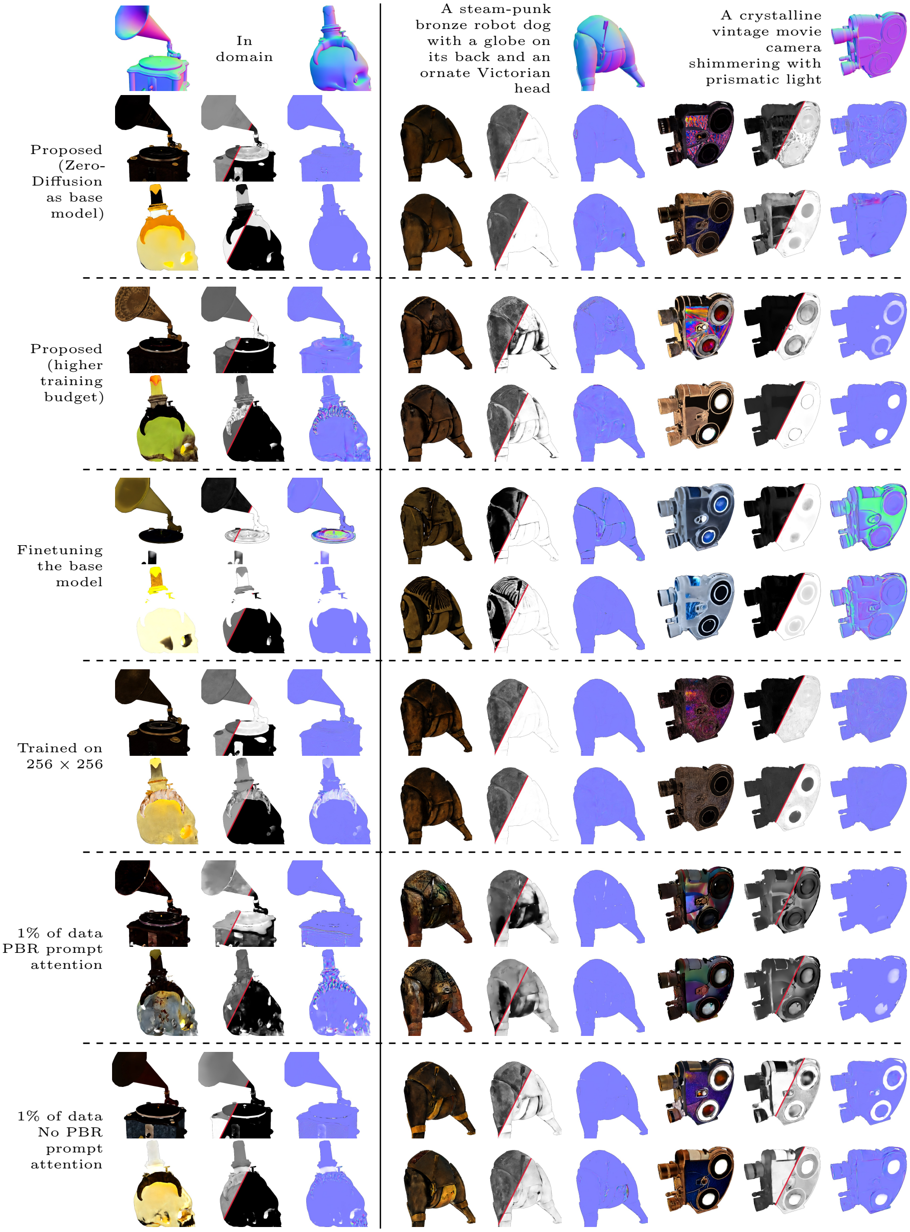

In domain
A steam-punk bronze robot dog with a globe on its back and an ornate Victorian head
A crystalline vintage movie camera shimmering with prismatic light
Proposed (Zero-Diffusion as base model)
Proposed (higher training budget)
Finetuning the base model
Trained on 256 × 256
1% of data PBR prompt attention
1% of data No PBR prompt attention

Fig. 8: Generated albedo, roughness/metallic and bump map images from the ablation studies. While significant quality differences are visible, only the finetuning approach and the data-sparse regime *with* PBR prompt cross-attention fail completely. The version that was trained on a smaller resolution does not break but does not result in maximum detail either. Best viewed digitally.

			2% held-out evaluation data								OOD					
			CMMD ↓		FID ↓		QAlign ↑				QAlign ↑				CLIPScore ↑	
							Ae		Q		Ae		Q			
			PBR	Relit	PBR	Relit	Albedo	Relit	Albedo	Relit	Albedo	Relit	Albedo	Relit	Albedo	Relit
Communication		one-way	16.44	13.38	20.90	16.39	1.95	1.97	2.37	2.48	1.91	1.59	2.35	1.69	17.57	17.54
		clockwise	6.78	2.76	12.21	11.53	2.04	2.02	2.63	2.64	**2.14**	1.70	2.77	**1.74**	**17.66**	17.54
		bi-directional	**6.30**	**1.79**	**11.65**	**10.64**	**2.11**	2.03	**2.75**	**2.66**	2.12	1.76	**2.78**	1.73	17.64	17.54
		Pixel-wise zero-conv	6.30	**1.79**	11.65	**10.64**	**2.11**	2.03	**2.75**	**2.66**	2.12	1.76	2.78	1.73	17.64	17.54
		Pixel-wise MLP	**5.43**	1.87	**11.43**	10.67	2.10	2.02	2.74	**2.66**	**2.26**	1.75	**2.96**	**1.81**	**17.79**	17.54
		Global Attention	7.60	5.22	13.61	11.93	1.94	1.98	2.51	2.60	1.99	1.72	2.71	1.80	17.66	17.54
		Collaborative Control	6.30	**1.79**	11.65	**10.64**	**2.11**	2.03	**2.75**	**2.66**	2.12	1.76	2.78	1.73	17.64	17.54
		Finetuning (with RGB output)	13.40	2.78	14.42	10.79	2.05	2.02	2.60	2.61	2.10	1.76	2.62	**1.86**	**17.78**	17.54
		Finetuning (without RGB output)	**5.25**	2.88	**11.41**	11.37	2.03	1.99	2.58	2.58	**2.26**	1.71	**2.97**	1.81	17.52	17.54
		PBR VAE	**6.30**	**1.79**	**11.65**	**10.64**	2.11	**2.03**	**2.75**	**2.66**	2.12	1.76	2.78	1.73	**17.64**	17.54
		RGB VAE on triplets	84.66	5.99	25.81	11.63	**2.16**	1.99	2.67	2.55	**2.30**	1.71	**2.95**	**1.80**	17.59	17.54
Training budget		**1 A100, two days**	6.30	1.79	11.65	10.64	**2.11**	2.03	**2.75**	2.66	**2.12**	1.76	2.78	1.73	17.64	17.54
		8 A100s, two days	**2.96**	**1.12**	**9.55**	**9.76**	2.08	2.04	2.68	2.67	2.01	1.73	**2.82**	**1.82**	**17.71**	17.54
Training resolution		256×256	**2.23**	**1.44**	**9.82**	**10.20**	2.10	2.04	2.73	2.68	2.20	1.78	**3.13**	**1.80**	**17.74**	17.54
		512×512	6.30	1.79	11.65	10.64	**2.11**	2.03	**2.75**	2.66	**2.12**	1.76	2.78	1.73	17.64	17.54
Training data	**No PBR prompt attention**	1%	6.25	**1.43**	11.87	10.79	2.18	2.04	**2.86**	2.69	**2.35**	1.76	**3.35**	1.89	17.44	17.54
		5%	**5.77**	1.45	11.49	**10.54**	2.13	2.04	2.78	2.69	2.09	1.73	3.01	1.84	17.54	17.54
		20%	5.97	1.68	**11.50**	10.61	2.12	2.03	2.78	2.67	2.23	1.76	3.23	1.88	17.63	17.54
		98%	6.30	1.79	11.65	10.64	2.11	2.03	2.75	2.66	2.12	1.76	2.78	1.73	17.64	17.54
	PBR prompt attention	1%	20.61	4.25	18.35	12.16	**2.19**	2.03	2.76	2.62	2.25	1.61	2.83	1.80	17.55	17.54
		5%	12.17	2.58	14.95	10.97	2.13	2.04	2.71	2.65	2.27	1.80	2.97	**1.98**	17.59	17.54
		20%	11.35	2.33	14.78	10.78	2.13	2.03	2.74	2.65	2.29	1.76	3.07	1.77	**17.83**	17.54
		98%	9.18	2.57	13.25	11.02	2.10	2.03	2.68	2.64	2.17	1.80	2.84	1.87	17.38	17.54

Table 1: Quantative results for all evaluated variants. The ablation baseline is highlighted in bold, duplicated for easier comparisons within the individual ablations.

How much data does the model need to train? Finally, we evaluate the performance of Collaborative Control when trained on decreasingly smaller amounts of data. For this purpose, we evaluate models trained on 98%, 20%, 5% and 1% of the 6M training images in our full dataset, showing in both the quantitative and qualitative results that the proposed approach is very data efficient and performing well even when trained on only a few thousand images. We train these models both with and without text cross-attention in the PBR model: crucially, we observe that it is necessary to disable the text cross-attention layer in the PBR model, and that this effect gets more pronounced with less data. We hypothesize that the model overfits to the training data.

5.2 Compatibility with other control techniques

As a closing experiment, we illustrate that Collaborative Control is compatible with other control techniques [11,43,73], which drastically expands the practical applications of the method. We demonstrate this specifically with IP-Adapter [73]. IP-Adapter allows us to condition the final output on a style image, by only introducing additional style cross-attention layers to the base model. We can apply an IP-Adapter overlay to the base model without requiring retraining, as illustrated in Fig. 9.

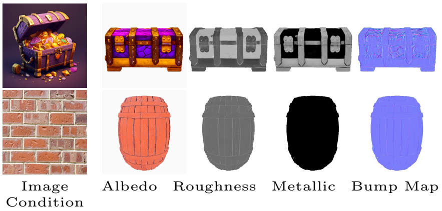


Fig. 9: Our PBR diffusion remains compatible with control techniques trained for the frozen RGB model they are linked to. We illustrate this using StableDiffusion 1.5 as the base model using the publically available IP-Adapter [73].

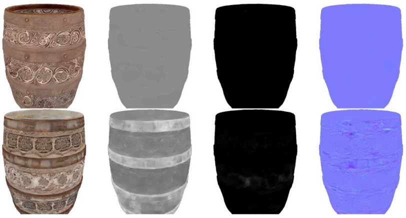


Fig. 10: The most common failure case of our proposed model is the absence of content in the roughness, metallic and bump maps. Prompting a *porcelain barrel with intricate designs* for two different random seeds illustrates this behaviour.

5.3 Limitations and failure cases

We identify two major failure cases: lack of detail in the roughness, metallic and bump maps, and a failure to follow OOD prompts. We attribute the former to the training data: Objaverse contains many objects with constant roughness and metallic properties, and without any details in the surface bump map. This likely biases the model to such outputs, as shown illustrated in Fig. 10. Anecdotically, we have found that selecting a different random seed will often succeed where the first generation failed — practically, the model produces very diverse results even for the same prompt and the same conditioning geometry, so that we argue that this is not a significant issue. A failure to follow out-of-distribution prompts mostly when prompting structural features in the prompt are incompatible with the conditioning geometry, such as for example prompting *a gilded lion* for a table mesh. We hypothesize that the control signal from the PBR model conflicts with the text cross-attention in the frozen RGB model, resulting in lackluster outputs, although different random seeds occasionally resolve this issue, too.

6 Conclusion

In this work, we have proposed Collaborative Control, a new paradigm for leveraging a pre-trained image-based RGB diffusion model for generating high-quality PBR image content conditioned on object geometry. We have shown that this bi-directional control paradigm is extremely data efficient while retaining the high quality and expressiveness of the base RGB model, even when faced with text queries completely out of distribution for the PBR training data. The plug-and-play nature of our proposed approach is compatible with existing adaptations of the base RGB model, which we have illustrated with IP-Adapter for style guidance of the PBR content. The availability of high-quality PBR content generation as offered by our proposed approach opens up new avenues for graphics applications, specifically in Text-to-Texture.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. stat **1050**, 21 (2016)
3. Burley, B.: Physically based shading at disney. In: ACM Transactions on Graphics (SIGGRAPH) (2012)
4. Cao, T., Kreis, K., Fidler, S., Sharp, N., Yin, K.: Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4169–4181 (2023)
5. Chambon, T., Heitz, E., Belcour, L.: Passing multi-channel material textures to a 3-channel loss. In: ACM SIGGRAPH 2021 Talks, pp. 1–2 (2021)
6. Chen, D.Z., Siddiqui, Y., Lee, H.Y., Tulyakov, S., Nießner, M.: Text2tex: Text-driven texture synthesis via diffusion models. arXiv preprint arXiv:2303.11396 (2023)
7. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. arXiv preprint arXiv:2303.13873 (2023)
8. Chung, J., Lee, S., Nam, H., Lee, J., Lee, K.M.: Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. arXiv preprint arXiv:2311.13384 (2023)
9. Cook, R.L., Torrance, K.E.: A reflectance model for computer graphics. ACM Transactions on Graphics (ToG) (1982)
10. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13142–13153 (2023)
11. Denis Zavadski, J.F.F., Rother, C.: Controlnet-xs: Designing an efficient and effective architecture for controlling text-to-image diffusion models (2023)
12. Duan, Y., Guo, X., Zhu, Z.: Diffusiondepth: Diffusion denoising approach for monocular depth estimation. arXiv preprint arXiv:2303.05021 (2023)
13. Foong, T.Y., Kotyan, S., Mao, P.Y., Vargas, D.V.: The challenges of image generation models in generating multi-component images. arXiv preprint arXiv:2311.13620 (2023)
14. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
15. Guo, P., Hao, H., Caccavale, A., Ren, Z., Zhang, E., Shan, Q., Sankar, A., Schwing, A.G., Colburn, A., Ma, F.: Stabledreamer: Taming noisy score distillation sampling for text-to-3d. arXiv preprint arXiv:2312.02189 (2023)
16. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. arXiv preprint arXiv:2104.08718 (2021)
17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
18. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
19. Hu, L., Gao, X., Zhang, P., Sun, K., Zhang, B., Bo, L.: Animate anyone: Consistent and controllable image-to-video synthesis for character animation. arXiv preprint arXiv:2311.17117 (2023)

20. Huang, T., Zeng, Y., Zhang, Z., Xu, W., Xu, H., Xu, S., Lau, R.W., Zuo, W.: Dreamcontrol: Control-based text-to-3d generation with 3d self-prior. arXiv preprint arXiv:2312.06439 (2023)
21. Huang, Y., Wang, J., Shi, Y., Qi, X., Zha, Z.J., Zhang, L.: Dreamtime: An improved optimization strategy for text-to-3d content creation. arXiv preprint arXiv:2306.12422 (2023)
22. Jayasumana, S., Ramalingam, S., Veit, A., Glasner, D., Chakrabarti, A., Kumar, S.: Rethinking fid: Towards a better evaluation metric for image generation. arXiv preprint arXiv:2401.09603 (2023)
23. Jin, Z., Shen, X., Li, B., Xue, X.: Training-free diffusion model adaptation for variable-sized text-to-image synthesis. arXiv preprint arXiv:2306.08645 (2023)
24. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
25. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. Advances in Neural Information Processing Systems **34**, 852–863 (2021)
26. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
27. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2020)
28. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. arXiv preprint arXiv:2312.02145 (2023)
29. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
30. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
31. Knodt, J., Gao, X.: Consistent mesh diffusion. arXiv preprint arXiv:2312.00971 (2023)
32. Le, C., Hetang, C., Cao, A., He, Y.: Euclidreamer: Fast and high-quality texturing for 3d models with stable diffusion depth. arXiv preprint arXiv:2311.15573 (2023)
33. Li, X., Zhang, Q., Kang, D., Cheng, W., Gao, Y., Zhang, J., Liang, Z., Liao, J., Cao, Y.P., Shan, Y.: Advances in 3d generation: A survey. arXiv preprint arXiv:2401.17807 (2024)
34. Lin, S., Liu, B., Li, J., Yang, X.: Common diffusion noise schedules and sample steps are flawed. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5404–5411 (2024)
35. Liu, F., Wu, D., Wei, Y., Rao, Y., Duan, Y.: Sherpa3d: Boosting high-fidelity text-to-3d generation via coarse 3d prior. arXiv preprint arXiv:2312.06655 (2023)
36. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023)
37. Liu, Y.T., Luo, G., Sun, H., Yin, W., Guo, Y.C., Zhang, S.H.: Pi3d: Efficient text-to-3d generation with pseudo-image diffusion. arXiv preprint arXiv:2312.09069 (2023)
38. Liu, Z., Li, Y., Lin, Y., Yu, X., Peng, S., Cao, Y.P., Qi, X., Huang, X., Liang, D., Ouyang, W.: Unidream: Unifying diffusion priors for relightable text-to-3d generation. arXiv preprint arXiv:2312.08754 (2023)

39. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3d: Single image to 3d using cross-domain diffusion. arXiv preprint arXiv:2310.15008 (2023)
40. Ma, B., Deng, H., Zhou, J., Liu, Y.S., Huang, T., Wang, X.: Geodream: Disentangling 2d and geometric priors for high-fidelity and consistent 3d generation. arXiv preprint arXiv:2311.17971 (2023)
41. Ma, Y., Fan, Y., Ji, J., Wang, H., Sun, X., Jiang, G., Shu, A., Ji, R.: X-dreamer: Creating high-quality 3d content by bridging the domain gap between text-to-2d and text-to-3d generation. arXiv preprint arXiv:2312.00085 (2023)
42. Mercier, A., Nakhli, R., Reddy, M., Yasarla, R., Cai, H., Porikli, F., Berger, G.: Hexagen3d: Stablediffusion is just one step away from fast and diverse text-to-3d generation. arXiv preprint arXiv:2401.07727 (2024)
43. Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)
44. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. Advances in neural information processing systems **29** (2016)
45. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
46. Raj, A., Kaza, S., Poole, B., Niemeyer, M., Ruiz, N., Mildenhall, B., Zada, S., Aberman, K., Rubinstein, M., Barron, J., et al.: Dreambooth3d: Subject-driven text-to-3d generation. arXiv preprint arXiv:2303.13508 (2023)
47. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
48. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. Advances in neural information processing systems **29** (2016)
49. Sarkar, A., Mai, H., Mahapatra, A., Lazebnik, S., Forsyth, D.A., Bhattad, A.: Shadows don't lie and lines can't bend! generative models don't know projective geometry... for now. arXiv preprint arXiv:2311.17138 (2023)
50. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. arXiv preprint arXiv:2311.17042 (2023)
51. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022)
52. Sharma, P., Jampani, V., Li, Y., Jia, X., Lagun, D., Durand, F., Freeman, W.T., Matthews, M.: Alchemist: Parametric control of material properties with diffusion models. arXiv preprint arXiv:2312.02970 (2023)
53. Shi, R., Chen, H., Zhang, Z., Liu, M., Xu, C., Wei, X., Chen, L., Zeng, C., Su, H.: Zero123++: a single image to consistent multi-view diffusion base model. arXiv preprint arXiv:2310.15110 (2023)
54. Shi, Y., Wang, P., Ye, J., Long, M., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. arXiv preprint arXiv:2308.16512 (2023)
55. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)
56. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)

57. Stan, G.B.M., Wofk, D., Fox, S., Redden, A., Saxton, W., Yu, J., Aflalo, E., Tseng, S.Y., Nonato, F., Muller, M., et al.: Ldm3d: Latent diffusion model for 3d. arXiv preprint arXiv:2305.10853 (2023)
58. Subias, J.D., Lagunas, M.: In-the-wild material appearance editing using perceptual attributes. In: Computer Graphics Forum. vol. 42, pp. 333–345. Wiley Online Library (2023)
59. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
60. Tang, B., Wang, J., Wu, Z., Zhang, L.: Stable score distillation for high-quality 3d generation. arXiv preprint arXiv:2312.09305 (2023)
61. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023)
62. https://huggingface.co/drhead: Huggingface zerodiffusion model weights v0.9. https://huggingface.co/drhead/ZeroDiffusion, accessed: 2024-02-08
63. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International conference on machine learning. pp. 1747–1756. PMLR (2016)
64. Vecchio, G., Martin, R., Roullier, A., Kaiser, A., Rouffet, R., Deschaintre, V., Boubekeur, T.: Controlmat: A controlled generative approach to material capture. arXiv preprint arXiv:2309.01700 (2023)
65. Wan, Z., Paschalidou, D., Huang, I., Liu, H., Shen, B., Xiang, X., Liao, J., Guibas, L.: Cad: Photorealistic 3d generation via adversarial distillation. arXiv preprint arXiv:2312.06663 (2023)
66. Wang, P., Fan, Z., Xu, D., Wang, D., Mohan, S., Iandola, F., Ranjan, R., Li, Y., Liu, Q., Wang, Z., et al.: Steindreamer: Variance reduction for text-to-3d score distillation via stein identity. arXiv preprint arXiv:2401.00604 (2023)
67. Wang, Z., Li, M., Chen, C.: Luciddreaming: Controllable object-centric 3d generation. arXiv preprint arXiv:2312.00588 (2023)
68. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. arXiv preprint arXiv:2305.16213 (2023)
69. Wu, H., Zhang, Z., Zhang, W., Chen, C., Liao, L., Li, C., Gao, Y., Wang, A., Zhang, E., Sun, W., et al.: Q-align: Teaching lmms for visual scoring via discrete text-defined levels. arXiv preprint arXiv:2312.17090 (2023)
70. Wu, T., Li, Z., Yang, S., Zhang, P., Pan, X., Wang, J., Lin, D., Liu, Z.: Hyperdreamer: Hyper-realistic 3d content generation and editing from a single image. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–10 (2023)
71. Wu, Z., Zhou, P., Yi, X., Yuan, X., Zhang, H.: Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. arXiv preprint arXiv:2401.09050 (2024)
72. Xu, X., Lyu, Z., Pan, X., Dai, B.: Matlaber: Material-aware text-to-3d via latent brdf auto-encoder. arXiv preprint arXiv:2308.09278 (2023)
73. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models (2023)
74. Yeh, Y.Y., Huang, J.B., Kim, C., Xiao, L., Nguyen-Phuoc, T., Khan, N., Zhang, C., Chandraker, M., Marshall, C.S., Dong, Z., et al.: Texturedreamer: Image-guided texture synthesis through geometry-aware diffusion. arXiv preprint arXiv:2401.09416 (2024)

75. Youwang, K., Oh, T.H., Pons-Moll, G.: Paint-it: Text-to-texture synthesis via deep convolutional texture map optimization and physically-based rendering. arXiv preprint arXiv:2312.11360 (2023)
76. Yu, K., Liu, J., Feng, M., Cui, M., Xie, X.: Boosting3d: High-fidelity image-to-3d by boosting 2d diffusion prior to 3d prior with progressive learning. arXiv preprint arXiv:2311.13617 (2023)
77. Zeng, X.: Paint3d: Paint anything 3d with lighting-less texture diffusion models. arXiv preprint arXiv:2312.13913 (2023)
78. Zhang, J., Tang, Z., Pang, Y., Cheng, X., Jin, P., Wei, Y., Yu, W., Ning, M., Yuan, L.: Repaint123: Fast and high-quality one image to 3d generation with progressive controllable 2d repainting. arXiv preprint arXiv:2312.13271 (2023)
79. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)
80. Zhao, W., Rao, Y., Liu, Z., Liu, B., Zhou, J., Lu, J.: Unleashing text-to-image diffusion models for visual perception. arXiv preprint arXiv:2303.02153 (2023)
81. Zhou, L., Shih, A., Meng, C., Ermon, S.: Dreampropeller: Supercharge text-to-3d generation with parallel sampling. arXiv preprint arXiv:2311.17082 (2023)
82. Zhuang, J., Wang, C., Lin, L., Liu, L., Li, G.: Dreameditor: Text-driven 3d scene editing with neural fields. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–10 (2023)