Generative Detail Enhancement for Physically Based Materials

SAEED HADADAN, University of Maryland, College Park, NVIDIA, USA
BENEDIKT BITTERLI, NVIDIA, USA
TIZIAN ZELTNER, NVIDIA, Switzerland
JAN NOVÁK, NVIDIA, Czech Republic
FABRICE ROUSSELLE, NVIDIA, Switzerland
JACOB MUNKBERG, NVIDIA, Sweden
JON HASSELGREN, NVIDIA, Sweden
BARTLOMIEJ WRONSKI, NVIDIA, USA
MATTHIAS ZWICKER, University of Maryland, College Park, USA

Fig. 1. We enhance material definitions of existing 3D assets (**a**) by applying effects specified by text prompts such as aging, weathering, etc. Conditioned on a set of renderings, we synthesize the corresponding visuals in 2D using a diffusion model building on multi-view visual prompting [Deng et al. 2024] (**b**). We improve the multi-view consistency of the generator using two key additions—view-correlated noise and attention biasing (**c**)—that enable succesful inverse-rendering of the visual enhancements back to the original material textures (**d**).

We present a tool for enhancing the detail of physically based materials using an off-the-shelf diffusion model and inverse rendering. Our goal is to increase the visual fidelity of existing materials by adding, for instance, signs of wear, aging, and weathering that are tedious to author. To obtain realistic appearance with minimal user effort, we leverage a generative image model trained on a large dataset of natural images. Given the geometry, UV mapping, and basic appearance of an object, we proceed as follows: We render multiple views of the object and use them, together with an appearance-defining text prompt, to condition a diffusion model. The generated details are then back-propagated from the enhanced images to the material parameters via inverse rendering. For inverse rendering to be successful, the generated appearance has to be consistent across all the images. We propose two priors to address the multi-view consistency of the diffusion model. First, we ensure that the noise that seeds the diffusion process is itself consistent across views by integrating it from a view-independent UV space. Second, we enforce spatial consistency by biasing the attention mechanism via a projective constraint so that pixels attend strongly to their corresponding pixel locations in other views. Our approach does not require any training or finetuning of the diffusion model, is agnostic to the used material model, and the enhanced material properties, i.e., 2D PBR textures, can be further edited by artists. We demonstrate prompt-based material edits exhibiting high levels of realism and detail. This project is available at https://generative-detail.github.io.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Texturing**; *Ray tracing*.

Additional Key Words and Phrases: Mutli-view consistent diffusion, Generative Graphics, Physically-based rendering, Inverse rendering

ACM Reference Format:
Saeed Hadadan, Benedikt Bitterli, Tizian Zeltner, Jan Novák, Fabrice Rousselle, Jacob Munkberg, Jon Hasselgren, Bartlomiej Wronski, and Matthias

Authors' addresses: Saeed Hadadan, University of Maryland, College Park, and NVIDIA, MD, 20740, USA, saeedhd@umd.edu; Benedikt Bitterli, NVIDIA, WA, USA, bbitterli@nvidia.com; Tizian Zeltner, NVIDIA, Switzerland, tzeltner@nvidia.com; Jan Novák, NVIDIA, Czech Republic, jnovak@nvidia.com; Fabrice Rousselle, NVIDIA, Switzerland, frousselle@nvidia.com; Jacob Munkberg, NVIDIA, Sweden, frousselle@nvidia.com; Jon Hasselgren, NVIDIA, Sweden, frousselle@nvidia.com; Bartlomiej Wronski, NVIDIA, USA, bwronski@nvidia.com; Matthias Zwicker, University of Maryland, College Park, MD, USA, zwicker@cs.umd.edu.

SIGGRAPH Conference Papers '25, August 10–14, 2025, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/2025/08
https://doi.org/10.1145/3721238.3730751

Zwicker. 2025. Generative Detail Enhancement for Physically Based Materials. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 26 pages. https://doi.org/10.1145/3721238.3730751

1 INTRODUCTION

Depicting rich 3D worlds is a driving goal of computer graphics. While achieving this goal is possible today for experienced artists with expert tools, the pareto principle applies: The creative step of authoring the overall look of an asset takes little time in comparison to the disproportionate effort of infusing details and imperfections of the real world. Our goal is therefore to create a tool that enhances 3D objects with appearance details requiring comparatively minimal effort from the artist.

For this, we turn to diffusion models [Ho et al. 2020] that are capable of producing realistic visuals and can be conditioned using text prompts and guiding images. A key consideration, however, is the amount of training data available. While datasets containing 3D objects and materials exist [Deitke et al. 2022; Vecchio and Deschaintre 2024], they cannot compete with natural image datasets in size and diversity, which directly impacts the model capabilities. We therefore build our tool using an off-the-shelf diffusion model that was trained on an internet-scale image set.

We combine the diffusion model with a physically based renderer to enable two key editing features: 1) specifying the initial look of the object, and 2) outputing a material representation that is complient with traditional authoring workflows. Our algorithm works as follows. We start by rendering the original 3D asset from multiple views. Then we condition the diffusion model on a concatenation of these views, and a text prompt describing the desired detail enhancements. Since our goal is to merely enhance the appearance, we propose a specific way of using two publicly available ControlNets [Zhang et al. 2023] to condition the model on the asset geometry and initial appearance. Finally, the differences between the original renderings and the diffusion-generated views are backpropagated to the material parameters via inverse rendering.

The main challenge of multi-view generation with diffusion models is the consistency of individual details in all relevant views. We address this challenge with two contributions. First, we seed the diffusion model with noise that is itself consistent across the views. We adopt the idea of integral noise [Chang et al. 2024] and project a common UV-space noise pattern into each view in a variance-preserving manner. Second, we bias the attention maps in the diffusion model to encourage pixels to attend to their corresponding locations in different views. We compute the correspondences by reprojecting points between the views using ray tracing operations.

Our approach has several benefits. It avoids the impractical creation of a new dataset and/or retraining a large diffusion model. We preserve the original geometry and artistic intent, while modifying the material texture maps according to the user's target prompts. Because we only enhance the input material, the inverse rendering is more likely to succeed than if starting the optimization from scratch. Lastly, the input and output of our model are in the form of a classical 3D representation (e.g, triangles, textures) and thus perfectly multi-view consistent. This also allows users to further edit the appearance, integrate it into larger scenes, and render with common renderers.

2 PRIOR WORK

We review the related prior work in the following categories:

Multi-view consistency via latents sharing. TexPainter [Zhang et al. 2024b] and SyncMVD [Liu et al. 2024] denoise multi-view latents, and correlate the views in a shared texture space. Tex4D [Bao et al. 2024] extends the idea to the temporal domain using a video diffusion model. These methods however do not easily generalize to view dependent PBR materials. Patashnik et al. [2024] and Pandey et al. [2023] instead operate on intermediate features of the network to enforce 3D consistent transformations in the output images.

Another approach is to reuse the diffusion model's input noise across views as proposed by [Chang et al. 2024; Daras et al. 2024]. Our work extends this idea by anchoring the noise field in UV space for a more robust handling of disocclusions.

Multi-view consistency via view correspondences. Cross-frame attention modules have been devised for known depth maps [Tang et al. 2023], poses [Cerkezi et al. 2023], or epipolar constraints [Kant et al. 2024]. These methods however require large-scale training. Our method exploits the known UV mapping, and similarly to SyncTweedies [Kim et al. 2024], is training-free.

Text-guided 3D generation. DreamFusion [2023] pioneered generation of 3D models using text-to-image diffusion and score distillation sampling (SDS). The method has been extended to various representations [Lin et al. 2023; Yi et al. 2023], with improved objective functions [Wang et al. 2023; Xu et al. 2023], sampling [Zhu and Zhuang 2023], and material decomposition [Chen et al. 2023a; Youwang et al. 2024]. We provide additional discussion on how our method relates to SDS in the supplementary document.

Early methods [Cao et al. 2023; Chen et al. 2023b; Richardson et al. 2023] suffer from over-blurring due to the lack of view consistency. Follow-up work improved this using spatial attention [Shi et al. 2023], video-models [Voleti et al. 2024; Wu et al. 2024], and tiled inputs [Deng et al. 2024]. We use the latter idea in our work.

FlashTex [Deng et al. 2024], DreamMat [Zhang et al. 2024a], and MaPa [Zhang et al. 2024c] specialize in material reconstruction for a known scene. They leverage known priors by training a controlnet [Zhang et al. 2023] from geometry buffers (e.g. depth and normal), and lighting rendered with known, constant, materials (e.g. fully diffuse and specular). This helps greatly with view dependent shading effects, and separating shadows from material albedo. Vecchio et al. [2024] train a generative model to directly synthesize material maps. All these methods are costly as they require training with specialized object/material datasets, which we avoid.

Image and appearance editing. Text-guided diffusion models are often applied to image editing while preserving the semantics of the source image. This is achieved by manipulating the self-attention layers [Tumanyan et al. 2023], often combined with DDIM inversion [Mokady et al. 2023; Parmar et al. 2023]. RGB↔X [Zeng et al.

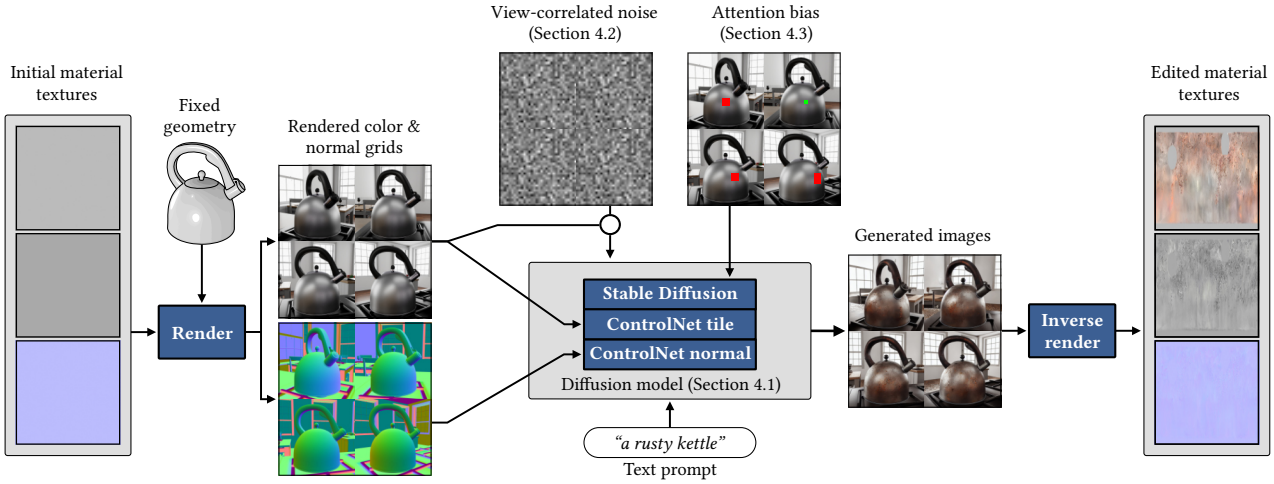


Fig. 2. Pipeline overview: Given a 3D asset including fixed geometry and initial material textures, we render color and normal images from multiple viewpoints (4 out of 16 views shown above). We then apply enhancements based on text prompts using a multi-view diffusion model designed to produce view-consistent outputs that edit the input images in a controllable manner. We achieve this by leveraging three distinct techniques, including suitable publicly available ControlNets, view-correlated noise, and cross-view attention bias. We finally obtain the edited material textures using inverse rendering.

2024] allows editing the content of images by first extracting irradiance and material maps, manually editing them, and then generating the corresponding realistic image. We consider a dual problem, where the inputs and outputs are PBR maps, and the intermediate step that enhances details involves generation of an image.

Material upscaling. Gauthier et al. [2024] consider a subset of detail enhancement, focusing on increasing the resolution of PBR material textures by inverse rendering upscaled images. However, they operate on flat-geometry and are therefore unable to synthesize detail in the context of the object geometry; this is one of our goals.

Video diffusion models. Video diffusion models [Blattmann et al. 2023; Hong et al. 2023; Yang et al. 2024] generate (view-consistent) video from text and image conditioning. Notably, SV3D [Voleti et al. 2024] adapts image-to-video diffusion model for novel multi-view synthesis and 3D generation. However, these models come at significant computational cost, and the generated frames are typically not as detailed as text-to-image models.

3 BACKGROUND

Diffusion models. Denoising diffusion leverages a bidirectional process where the forward pass gradually corrupts training data by iteratively adding Gaussian noise until the data becomes pure noise. The reverse process then learns to denoise the corrupted data through a neural network, which predicts and removes noise step-by-step. We use a latent space diffusion model [Rombach et al. 2022], which extracts the latent space using a variational autoencoder and performs the denoising steps with a U-Net architecture operating at different scales (see Figure 3 in Rombach et al. [2022]).

ControlNet. In order to guide the denoising process, the ControlNet model [Zhang et al. 2023] implements a dual-network architecture. The first network—a pretrained diffusion model—is locked down to perform the usual denoising task and cloned. The clone network is connected to the locked network using zero-initialized convolution layers, and enables precise spatial conditioning of the pretrained denoiser using images.

Attention. The attention operation [Vaswani et al. 2017] has been incorporated to diffusion models to capture relationships between different activations in the denoising process. The operation takes its inputs and embeds them in three learned linear spaces, referring to them as key, query, and value. Denoting the matrices of these embeddings $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, respectively the attention formula

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{1}$$

provides a mechanism for capturing the similarity between queries and keys (where d_k is the dimensionality of the key embedding).

A typical U-Net diffusion model features two types of attention: *cross-attention* layers that guide the denoising of image regions using a given text-prompt, and *self-attention* layers that allow regions within the image to influence each other. In the self-attention module, the $\mathbf{Q}\mathbf{K}^\top$ product forms a large attention score matrix, where entry $[i, j]$ describes how strongly region i attends to region j.

4 METHOD

Our method (see Figure 2) comprises three stages: forward rendering, detail generation, and inverse rendering discussed below. We then present our three technical contributions in Sections 4.1 to 4.3.

Forward rendering. We begin with a user-provided asset comprising of known geometry and the material to be enhanced, as well as a 3D scene providing context for the asset. We render the scene from a small number (9 to 16) of views in an orbit around the asset. The output of the renderer consists of the renderings as well as auxiliary buffers of surface normals, which serve to condition the diffusion model during the next stage.

Detail generation. The renderings from the previous stage are passed to an off-the-shelf diffusion model to add detail to the renderings, conditioned on a text prompt and the auxiliary buffers. Although the rendered views could be enhanced individually, we find that mutual view consistency is improved if we use *multi-view visual prompting* [Deng et al. 2024], in which we concatenate all views into a grid (e.g. 3×3 or 4×4) and enhance them simultaneously.

While multi-view prompting improves consistency across the views at a coarse scale, there remains enough variation in fine-scale detail between views to make reconstruction of highly detailed materials challenging. To remedy this, we propose two modifications to the diffusion model: using view-correlated input noise that is anchored in 3D space (Section 4.2) and biasing the attention layers with pixel-to-pixel correspondence information (Section 4.3).

Inverse rendering. We finally propagate the detail generated by the diffusion model back to the original material of the 3D asset, leveraging a differentiable renderer to minimize the difference between the enhanced views and the rendered material in a stochastic gradient optimization. In all our results, we optimize the spatially-varying albedo, normal, and roughness textures of a typical PBR material [Burley 2012], but any differentiable material definition can be used in principle. We initialize the optimization state with the original textures; this improves the convergence likelihood.

4.1 Structure-preserving detail enhancement

To enhance the rendered views while preserving the character of the input material, we follow the approach of Meng et al. [2022] and add a user-controlled amount of noise to the rendered views to get the initial state for diffusion. This is not enough, however, to preserve the original material and geometry. Hence, we additionally condition the diffusion model with two publicly available ControlNets [Zhang et al. 2023]: *ControlNet tile*, trained to do super-resolution, which we repurpose to respect the input view while enhancing details, and *ControlNet normal* that helps preserve lighting and curvature details using our auxiliary normal buffer as input.

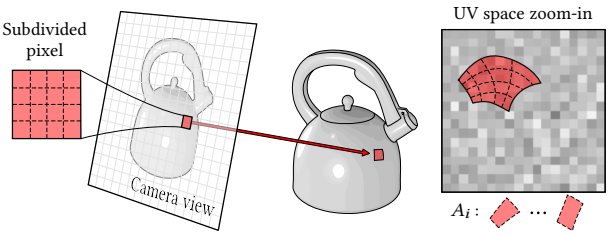
Together with our other modifications (Sec. 4.2 and 4.3) we find this to be effective at achieving consistency with the original material and across views, while avoiding expensive training of task-specific ControlNets as used in prior work [Deng et al. 2024].

4.2 View-correlated noise prior

Although the relationship between the initial noise input and the image output in diffusion is highly non-linear, the two are correlated. This has previously been exploited for temporal consistency by warping noise by a motion field, and we take a similar approach. Because diffusion models are highly sensitive to the noise statistics, the noise we produce must be uncorrelated within each view and have uniform variance, or we risk significant artifacts.

Based on Chang et al. [2024], we propose a simple method to correlate the initial noise of the diffusion model across views while preserving its statistics. In contrast to their application, we deal with a sparse set of views that do not undergo smooth motion. It would be challenging to warp an initial noise from a reference view due to the significant amount of disocclusion between views. Instead, we exploit the known geometry of the asset and anchor a reference noise field in the UV space of the asset.

For each view, we then project the noise from UV space (we use 1024×1024 noise textures) into image space and use this as the initial state for diffusion. Compared to the analytic integration of Chang et al. [2024], we use a simpler but effective *supersampling* approach. We subdivide each pixel into a grid of subpixels (4×4 in our implementation) and project the corners of each subpixel into the UV space of the object:



We then sample the noise field at the center of each projected subpixel, and compute an area-weighted average of the sampled noise values: $\sum_i f_i \cdot A_i$, where A_i is the area of each projected subpixel i, and f_i its noise value. Neighboring pixels don't overlap and generally average distinct sets of noise texels, and the resulting noise is independent within each view.

The variance of the projected noise is highly non-uniform, depending on the projected area of each pixel. To correct this, we could normalize the noise field by an estimate of its variance: $\sqrt{\sum_i A_i^2}$. However, because multiple subpixels may map to the same noise texel, we need to additionally account for the covariance between subpixels. We estimate this with $\mathrm{Cov}_i = \max(A_{\text{texel}}/A_i - 1, 0)$ where $A_{\text{texel}} = 1024^{-2}$ is the area of a noise texel. Intuitively, this counts how many times a distinct noise value is overcounted on average. The final normalization factor is then $\sqrt{\sum_i A_i^2 (1 + \mathrm{Cov}_i)}$. This matches the variance of projected- and reference noise.

In the case of extreme magnification of the noise texture, individual noise texels may project to multiple pixels and correlate noise within the image. This is rare and usually caused by missing or degenerate UVs, but it can negatively impact the quality of diffusion. As a safeguard, we smoothly blend the projected noise value with independent white noise when the pixel area in UV space, $\sum_i A_i$, approaches the area of a noise texel.

4.3 Pixel-correspondence attention bias

The second technique for improving the multi-view consistency amounts to biasing the self-attention mechanism of the diffusion model according to a reprojection prior.

As described in Section 3, the self-attention modules allow image regions to influence each other. We refer to these regions as *latent pixels* to emphasize that they map to pixels in the input/output image. The $\mathbf{QK}^\top$ product in the self-attention module forms a large $N \times N$ score matrix where entry $[i \in Q, j \in K]$ describes how strongly latent pixel i attends to latent pixel j; N is the total number of latent pixels.

Our goal is to increase the attention scores between latent pixels in different views that observe the *same surface patch*. This will increase

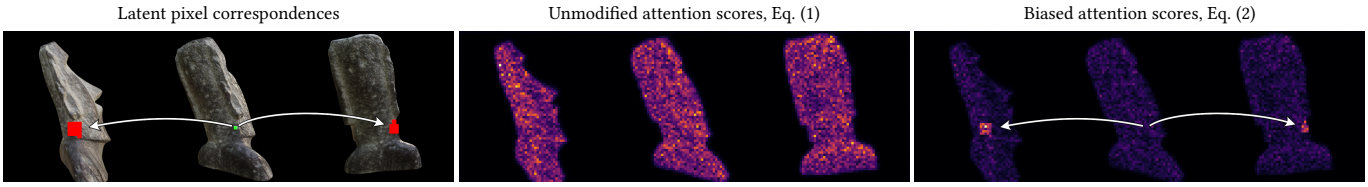


Fig. 3. Left: An example latent pixel (green) attends to corresponding image regions in other views (red). Middle: One *row* of the attention score matrix related to that green pixel is rearranged into a false-color image showing how much it attends to all other pixels in one stage of the diffusion model. Right: We bias the matrix elements in *columns* that correspond to the identified red regions to promote attention—and hence consistency—between these latents. Scores are visualized after the softmax in Eq. (1) and (2) and gamma-mapped for clarity. See the supplementary document for an extended version with 3×3 views.

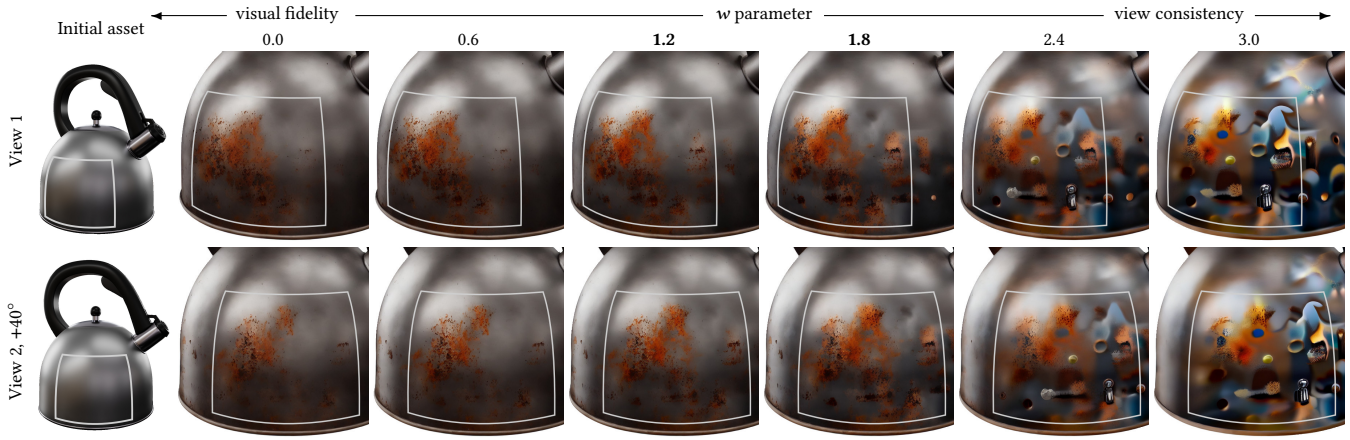
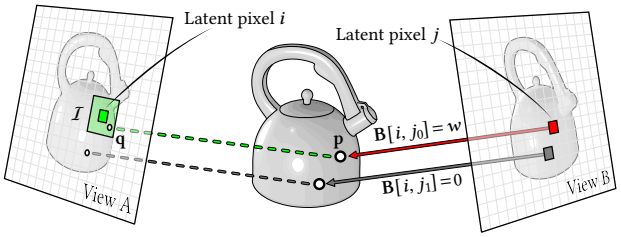


Fig. 4. The bias parameter w trades between visual fidelity (left insets) and multi-view consistency (right insets). The first column shows the initial asset in two views that condition the diffusion model. The white shapes outline a UV region, which we analyze in the insets to illustrate the impact of the w parameter on generated visuals. Values between 1.2 and 1.8 strike a good balance in this particular scene.

the chance that these latent pixels will denoise to consistent visuals in the resulting images. Conceptually, we construct an $N \times N$ *bias matrix* $\mathbf{B}$, where any positive value $\mathbf{B}[i, j]$ will boost the attention of pixel i to pixel j.

We determine the values of $\mathbf{B}$ as follows. For each pair of pixels i and j (where $i \neq j$) in the 3×3 latent image grid, we cast a ray through the center of latent pixel j into the 3D scene, finding the first hit point $\mathbf{p}$. We then project $\mathbf{p}$ onto the image plane containing pixel i and check that $\mathbf{p}$ and the projection $\mathbf{q}$ are mutually visible.



If the projection $\mathbf{q}$ is within a neighborhood $\mathcal{I}$ of pixel i, we set the value in the bias matrix to a user-defined constant: $\mathbf{B}[i, j] = w$.

The matrix is used to alter the attention operation as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top + \mathbf{B}}{\sqrt{d_k}}\right)\mathbf{V}. \qquad (2)$$

In practice, constructing the full $N \times N$ matrix is often prohibitive, and the bias term has to be evaluated on the fly (see Sec. 5).

The U-Net applies self-attention at multiple scales, and we adjust the size of neighbordhood $\mathcal{I}$ ($9^2, 5^2, 3^2, 1^2$ for layers 1-4, respectively) to map to the same size patch in the original image. Figure 3 visualizes exemplary attention scores before and after adding the bias for a single specific surface point. The increment w is a user-defined constant analysed in Figure 4. Increasing w improves view consistency but eventually generates less compelling appearance as the diffusion starts to lose its global view of the image.

5 IMPLEMENTATION

We implemented our system in PyTorch [Paszke et al. 2019] using publicly available models: the *tile* [Liylasviel 2025b] and *normal* [Liylasviel 2025a] variants of ControlNet (for color and normal inputs respectively) and the corresponding Hugging Face implementation [Wolf et al. 2020] of Stable Diffusion 1.5. We combine the two ControlNets by summing up their outputs[1].

We use Mitsuba 3 [Jakob et al. 2022] for GPU accelerated (differentiable) rendering. During inverse rendering, we apply a tonemapping operator [Reinhard et al. 2002] to the high-dynamic range renderings before comparing them with the (low-dynamic range)

[1]https://huggingface.co/docs/diffusers/en/using-diffusers/controlnet#multicontrolnet

Table 1. Runtime on NVIDIA RTX 5880 and memory usage for different numbers of conditioning views and resolutions. Implementing our attention biasing with xFormers (xF) instead of FlexAttention (FA) runs roughly 2× faster, but exceeds available memory with 9 and more views at 1024 × 1024.

		4 views		9 views		16 views	
		512^2	1024^2	512^2	1024^2	512^2	1024^2
FA	Runtime (s)	17	187	67	923	187	2816
	Peak memory (GB)	6.2	9.2	7.6	15.2	9.7	20.4
xF	Runtime (s)	8	102	34	-	100	-
	Peak memory (GB)	6.9	33.2	13.8	>45	33.2	>45

diffusion-generated target images using a relative L2 loss. It is important not to clip high values in order to preserve smooth specular highlights and avoid zero-valued gradients during backpropagation.

The ControlNet occasionally fails to preserve the exact silhouette of the rendered 3D geometry causing some background pixels to "bleed into the object" during the diffusion. We therefore stop gradient propagation for pixels that are within a fixed distance of the (precomputed) object boundary and downscale the loss at grazing incident angles based on a cosine-factor. Masked points still receive coverage from other views and are not removed from optimization.

Memory considerations and scalability. Our attention biasing can easily exhaust available GPU memory when operating on large images. While the pixel-to-pixel correspondences can easily be precomputed for a given asset (e.g. by storing matching pixel coordinates between each pair of views) we cannot afford to explicitly store the resulting $N \times N$ bias matrix in memory. This limits our choice of the attention framework. From the ones we tested [Dao 2024; Dao et al. 2022; Lefaudeux et al. 2022; Liu et al. 2021; Rabe and Staats 2021] only the recent FlexAttention [Dong et al. 2024] allowed us to scale to 16 views at resolution 1024^2 as it can apply attention biases on the fly. Table 1 illustrates how memory consumption and runtime performance scale with varying number of views and resolutions.

6 EXPERIMENTS

Comparisons. Our goal of enhancing given 3D assets with existing materials is distinct from recent work leveraging image models for view-consistent editing and generation. We first evaluate whether existing works can address our problem. Figure 5 justifies our technique by comparing it to related methods applied in our problem setting. The top half shows (multi-view) image generators, including SPAD [Kant et al. 2024], Diffusion Handles [Pandey et al. 2023], and RGB↔X [Zeng et al. 2024]. Because SPAD and Diffusion Handles are not designed to work with the given 3D geometry of an input asset, they struggle to render the asset accurately from multiple viewpoints. On the other hand, RGB↔X takes scene intrinsics as input, but it is not equipped to ensure multi-view consistency. We evaluated RGB↔X in the sequence RGB→X→RGB, where our initial renderings are inputs and the edited renderings are outputs.

The bottom part of Figure 5 compares our approach to material/texture generators given 3D geometry including DreamMat [Zhang et al. 2024a], Paint-it [Youwang et al. 2024], and TexPainter [Zhang et al. 2024b]. While they focus on material generation from scratch, our primary goal is to enhance existing materials. Hence our result is more faithful to the initial asset provided as input (shown in Figure 4), while also providing more realistic fine grained details. In principle, these approaches could be altered to enhance existing materials, e.g., by initializing their optimizable parameters with a given input textures. However, modifying prior works to suit our application would not improve their native performance.

We provide more examples and the corresponding recovered material attributes in the supplementary document.

Visual results. Figure 10 shows results of our complete pipeline, starting with basic assets, all the way to the recovered material parameters and the corresponding renderings. The Air Conditioner result in particular highlights the advantage of using a diffusion model trained on natural images. The rusting of the blades is distinct from that of the enclosure itself, which aligns with the expectation these components would age differently. In Figure 6 we show how classifier free guidance [Ho and Salimans 2022] enables the user to control the magnitude of the detail enhancements. We show further examples in the supplementary document and video.

Ablation. We validate our contributions and algorithmic choices to achieve view consistency of a pure 2D generative diffusion model by performing an ablation study presented in Figure 8. We modify the appearance and detail of the 3D assets Briefcase and Greek Vase: Pure ControlNet tile is able to modify the appearance of the object with respect to the initial material, but fails to achieve view consistency. Using view-correlated noise (Section. 4.2) we enhance the detail presence between different camera views, but some of the detail remains misaligned. Finally, our full model that biases attention (Section. 4.3) further increases the consistency.

View consistency and inverse rendering. Intuitively, view consistency between produced outputs is necessary to successfully reconstruct the material maps through inverse rendering. When different views of the same surface *disagree* and present different details, the inverse rendering process produces superimposed results or fails to converge. We present this effect in Figure 9.

7 LIMITATIONS AND FUTURE WORK

Additional consistency improvements. Although our contributions improve multi-view consistency, occasional deviations between views still occur at small scale. The inverse rendering typically resolves them by superimposing the conflicting visuals in the material maps. The final representation is view consistent, by design, but high-frequency view-dependent effects (e.g., mirror reflections) may end up baked in the albedo texture; see Figure 7. Since multi-view consistency of diffusion generators is an actively sought property, we believe future work will alleviate this either in a data-driven fashion, e.g., by employing multi-step optimization and video models, or by imposing additional geometric priors, such as identifying pixel correspondences using manifold walks on specular surfaces [Jakob and Marschner 2012].

Controlling the diffusion model. We currently expose only a text prompt as the control over the generated detail. Future work could explore more fine-grained user control, such as CLIP priors [Face

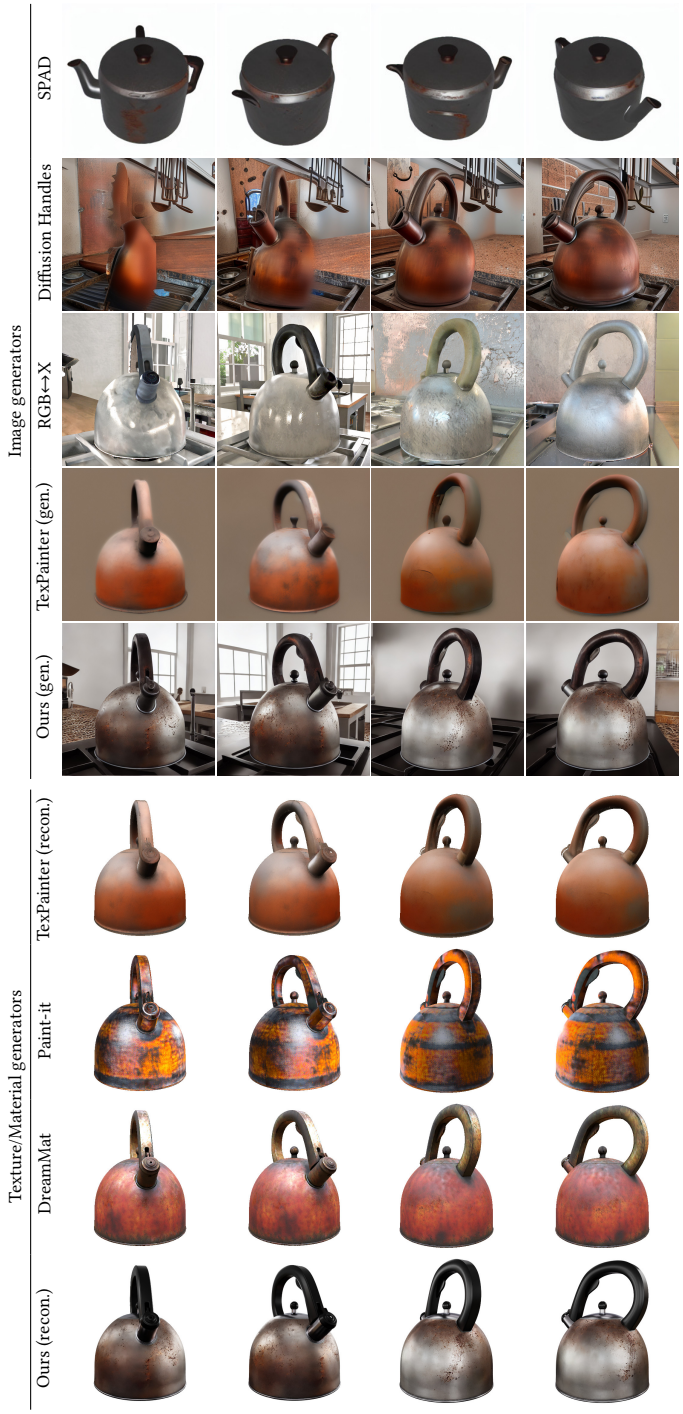


Fig. 5. Prior work comparison on KETTLE with the prompt *"Rusty scratched kettle"*. Image generators such as SPAD and Diffusion Handles are not designed to reproduce an input gometry. RGB↔X takes scene intrinsics as input, but is not equipped with multi-view consistency. Our problem is more related to material generation techniques. DreamMat and Paint-it use variants of SDS, which tends to output blurrier results. TexPainter does not allow for view-dependent effects. These techniques generate the output material from scratch while we enhance an input material. Both our approach and TexPainter do multi-view generation + reconstruction, shown in both segments above.

2025; Ramesh et al. 2022] or manipulations using texture exemplars [Guerrero-Viu et al. 2024]. The recently published Stable Diffusion 3.x [Esser et al. 2024; Stability AI 2025] supports more sophisticated text encoders and ControlNets, and swapping them in place of our current diffusion model provides a near-term avenue for better control. Importantly, artists still retain full editability of the material produced by our tool using traditional workflows.

Enhancing macro geometry. We focused on visual enhancements that can be captured using texture maps without attempting to refine the input macro geometry. While inverse rendering is in principle capable of updating the mesh, we leave this for future work.

Manual hyperparameter tuning. The attention bias w is currently a manually tuned hyperparameter. Although the range of reasonable values is limited ($[0 - 3.5]$ in our experiments), and w can be tuned quickly on low-resolution images, a parameter-free biasing would make the technique more practical.

8 CONCLUSION

We presented a method for enhancing the detail of a classically authored material using a diffusion model. Our method renders the provided material from multiple views, adds details to the renderings using a diffusion model, and then backpropagates the changes to the material using inverse rendering. Inverse rendering requires detail to be consistent across views, and we achieve this with two technical contributions: noise correlation by projecting from a reference noise anchored in the UV space, and attention biasing using the known geometry of the object. This requires no new datasets or expensive retraining and is largely built from off-the-shelf, pre-trained components.

The resulting method serves the important use case of *human-in-the-loop* authoring: Rather than entirely replacing the artist and generating materials from scratch, we allow the artist to maintain creative control using traditional workflows, while reducing the time spent on tedious detailing of assets—analogous to "auto-complete" for material detail. Because the input and output of our method are traditional materials, our method can be used at any stage in the authoring process, and the produced enhacements arbitrarily post-processed, blended, and combined. We believe our work builds a solid foundation for future practical tools that will further improve the robustness and controllability of the generative process.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. IIS2126407. We would like to thank Aaron Lefohn for supporting this research, and NVIDIA for funding the work with an NVIDIA academic partnership. We would also like to thank Arash Vahdat and Weili Nie for the highly insightful discussions on multi-view generation using diffusion models.

REFERENCES

Jingzhi Bao, Xueting Li, and Ming-Hsuan Yang. 2024. Tex4D: Zero-shot 4D Scene Texturing with Video Diffusion Models. *arXiv preprint arxiv:2410.10821* (2024).

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. 2023. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. *arXiv:2311.15127* (2023).

Fig. 6. Controlling the magnitude of detail enhancements using classifier free guidance. The scaling factor allows the user to apply more conservative changes and stay closer to the input or obtain more pronounced changes, as desired.

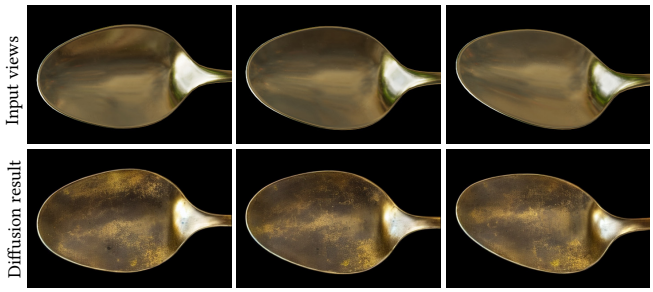


Fig. 7. Environment reflections on the spoon (top) are enhanced with diffuse details that follow the reflections across views (bottom) in a manner that may be inconsistent with user intentions.

Brent Burley. 2012. Physically-Based Shading at Disney. SIGGRAPH 2012 Course: Physically-Based Shading.

Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. 2023. TexFusion: Synthesizing 3D Textures with Text-Guided Image Diffusion Models . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 4146–4158.

Llukman Cerkezi, Aram Davtyan, Sepehr Sameni, and Paolo Favaro. 2023. Multi-View Unsupervised Image Generation with Cross Attention Guidance. arXiv:2312.04337 https://arxiv.org/abs/2312.04337

Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C. Azevedo. 2024. How I Warped Your Noise: a Temporally-Correlated Noise Prior for Diffusion Models. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=pzElnMrgSD

Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Niesner. 2023b. Text2Tex: Text-driven Texture Synthesis via Diffusion Models . In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. 18512–18522.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023a. Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 22246–22256.

Tri Dao. 2024. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *International Conference on Learning Representations (ICLR)*.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Giannis Daras, Weili Nie, Karsten Kreis, Alex Dimakis, Morteza Mardani, Nikola Borislavov Kovachki, and Arash Vahdat. 2024. Warped diffusion: Solving video inverse problems with image diffusion models. *arXiv preprint arXiv:2410.16152* (2024).

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2022. Objaverse: A Universe of Annotated 3D Objects. arXiv:2212.08051 [cs.CV] https://arxiv.org/abs/2212.08051

Kangle Deng, Timothy Omernick, Alexander Weiss, Deva Ramanan, Jun-Yan Zhu, Tinghui Zhou, and Maneesh Agrawala. 2024. FlashTex: Fast Relightable Mesh Texturing with LightControlNet. arXiv:2402.13251 [cs.GR] https://arxiv.org/abs/2402.13251

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. 2024. Flex Attention: A Programming Model for Generating Optimized Attention Kernels. arXiv:2412.05496 [cs.LG] https://arxiv.org/abs/2412.05496

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. 2024. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. arXiv:2403.03206 [cs.CV] https://arxiv.org/abs/2403.03206

Hugging Face. 2025. Stable UnCLIP Pipeline Documentation. https://huggingface.co/docs/diffusers/en/api/pipelines/stable_unclip. Accessed: 2025-01-15.

Alban Gauthier, Bernhard Kerbl, Jérémy Levallois, Robin Faury, Jean-Marc Thiery, and Tamy Boubekeur. 2024. MatUp: Repurposing Image Upsamplers for SVBRDFs. *Computer Graphics Forum* 43, 4 (2024).

Julia Guerrero-Viu, Milos Hasan, Arthur Roullier, Midhun Harikumar, Yiwei Hu, Paul Guerrero, Diego Gutiérrez, Belen Masia, and Valentin Deschaintre. 2024. TexSliders: Diffusion-Based Texture Editing in CLIP Space. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH '24)*. ACM, 1–11. https://doi.org/10.1145/3641519.3657444

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 6840–6851. https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

Jonathan Ho and Tim Salimans. 2022. Classifier-Free Diffusion Guidance. arXiv:2207.12598 [cs.LG] https://arxiv.org/abs/2207.12598

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. 2023. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers. In *The Eleventh International Conference on Learning Representations*.

Wenzel Jakob and Steve Marschner. 2012. Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 31, 4 (July 2012), 58:1–58:13. https://doi.org/10.1145/2185520.2185554

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. 2022. *Mitsuba 3 renderer*. https://mitsuba-renderer.org.

Yash Kant, Ziyi Wu, Michael Vasilkovsky, Guocheng Qian, Jian Ren, Riza Alp Guler, Bernard Ghanem, Sergey Tulyakov, Igor Gilitschenski, and Aliaksandr Siarohin. 2024. SPAD : Spatially Aware Multiview Diffusers. arXiv:2402.05235 [cs.CV] https://arxiv.org/abs/2402.05235

Jaihoon Kim, Juil Koo, Kyeongmin Yeo, and Minhyuk Sung. 2024. SyncTweedies: A General Generative Framework Based on Synchronized Diffusions. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 95198–95237. https://proceedings.neurips.cc/paper_files/paper/2024/file/ad1efab57a04d93f097e7fbb2d4fc054-Paper-Conference.pdf

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. 2022. xFormers: A modular and hackable Transformer modelling library. https://github.com/facebookresearch/xformers.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3D: High-Resolution Text-to-3D Content Creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. 2024. Text-Guided Texturing by Synchronized Multi-View Diffusion. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) *(SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 60, 11 pages. https://doi.org/10.1145/3680528.3687621

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. 2021. Swin Transformer V2: Scaling Up Capacity and Resolution. arXiv:2111.09883 [cs.CV]

Liylasviel. 2025a. ControlNet NormalBae Model (v1.1p, SD15). https://huggingface.co/lllyasviel/control_v11p_sd15_normalbae. Accessed: 2025-01-15.

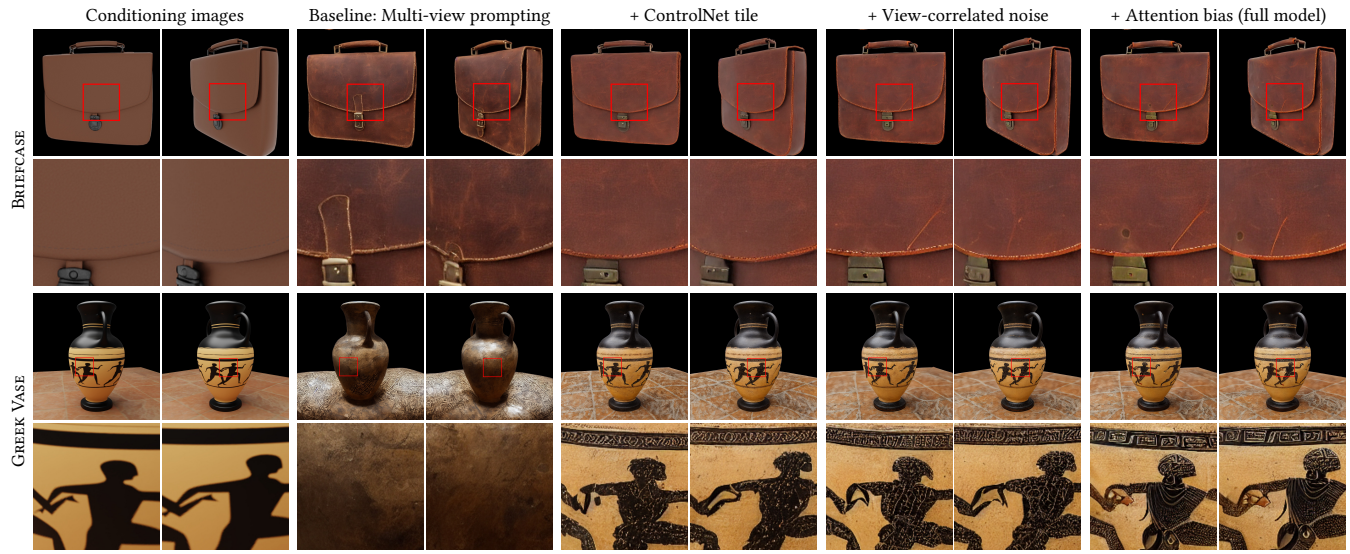


Fig. 8. Impact of individual components of our method, added one-by-one to the baseline. The ControlNet tile provides conditioning on initial images. The view-correlated noise and biased attention improve multi-view consistency. The supplementary includes a variant where images are warped to a single view.

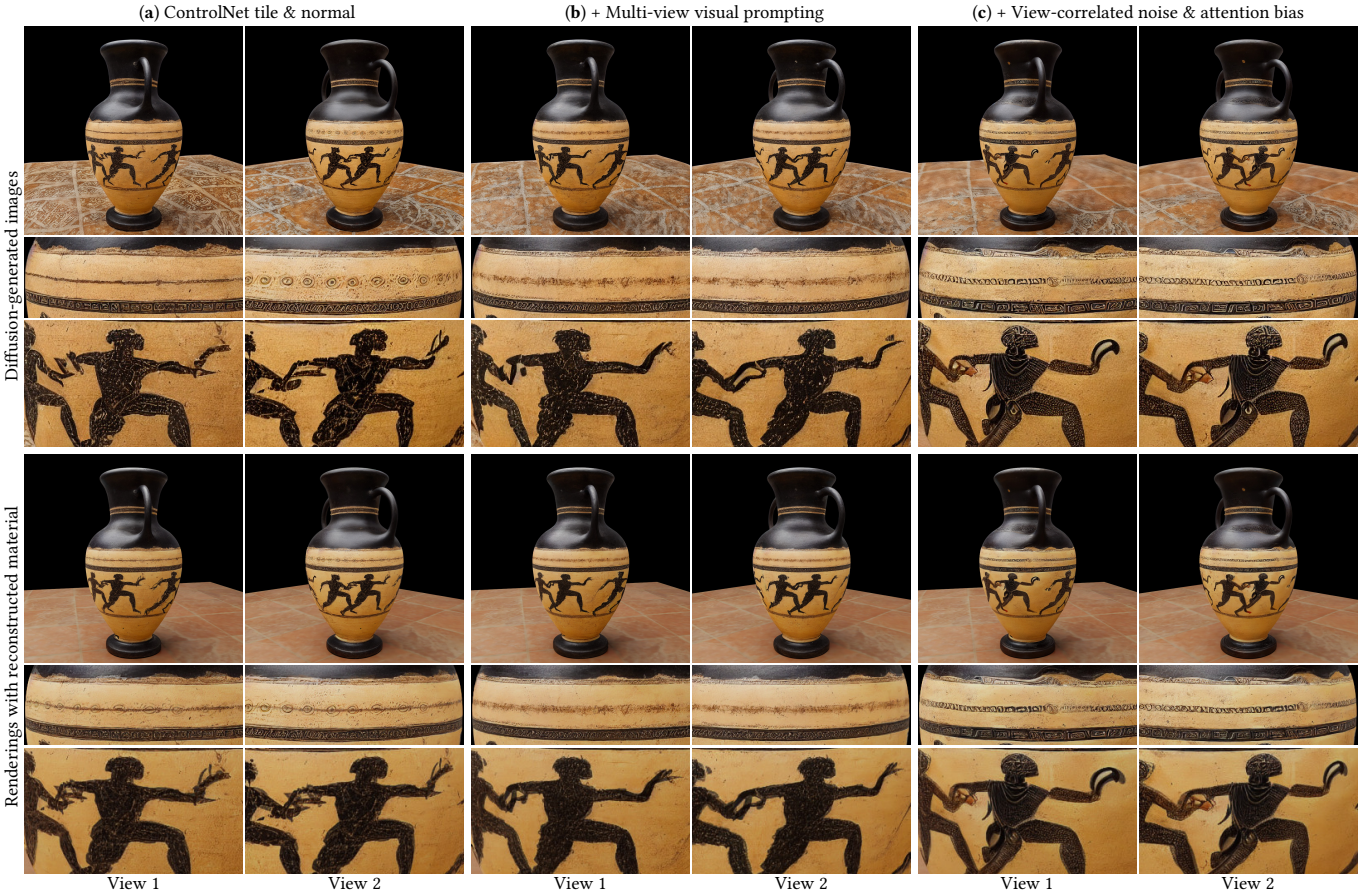


Fig. 9. Ablation of our model, first without the multi-view visual prompting (i.e., assembling the conditioning images into a grid) [Deng et al. 2024] (**a**), with it (**b**), and finally with our two techniques for improving multi-view consistency. The supplementary includes a variant where images are warped to a single view.

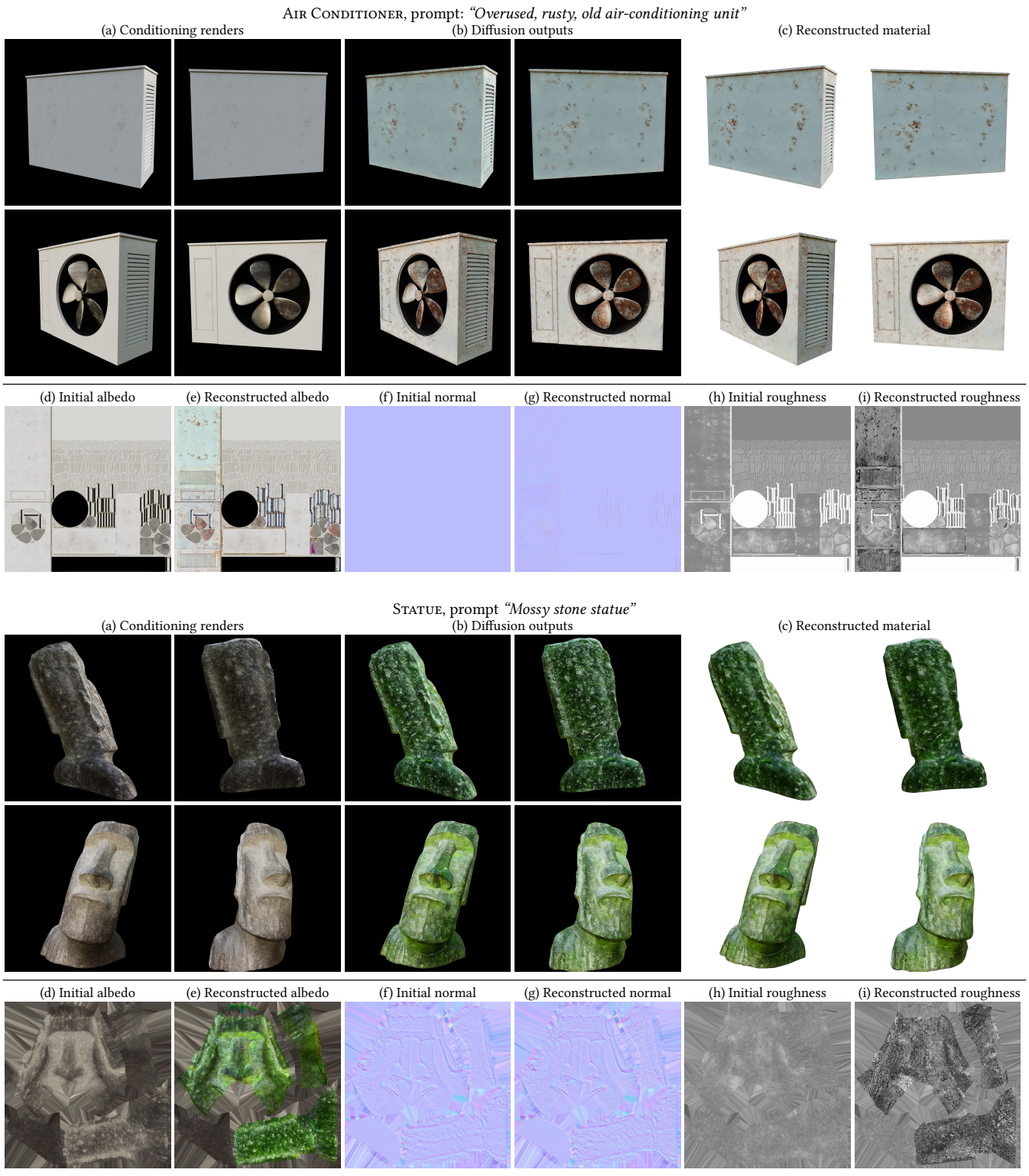


Fig. 10. We use renderings of the original asset (two pairs of two adjacent views are shown in (a)) to condition the diffusion model to produce images with enhanced appearance (b), which is then backpropagated into the original material definition. In (c), we show the resulting asset on four frames from a turntable animation (we picked frames that correspond to the conditioning views); see the accompanying video for the full animation. Columns (d) to (i) show albedo, normal, and roughness textures of the initial and reconstructed material.

Liylasviel. 2025b. ControlNet Tile Model (v1.1f1e, SD15). https://huggingface.co/lllyasviel/control_v11f1e_sd15_tile. Accessed: 2025-01-15.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2022. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=aBsCjcPu_tE

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2023. Null-text Inversion for Editing Real Images using Guided Diffusion Models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6038–6047.

Karran Pandey, Paul Guerrero, Matheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy Mitra. 2023. Diffusion Handles: Enabling 3D Edits for Diffusion Models by Lifting Activations to 3D. arXiv:2312.02190 [cs.CV] https://arxiv.org/abs/2312.02190

Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. 2023. Zero-shot Image-to-Image Translation. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*. Article 11, 11 pages.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG] https://arxiv.org/abs/1912.01703

Or Patashnik, Rinon Gal, Daniel Cohen-Or, Jun-Yan Zhu, and Fernando De La Torre. 2024. Consolidating Attention Features for Multi-view Image Editing. In *SIGGRAPH Asia 2024 Conference Papers (SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 40, 12 pages. https://doi.org/10.1145/3680528.3687611

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh International Conference on Learning Representations*.

Markus N. Rabe and Charles Staats. 2021. Self-attention Does Not Need $O(n^2)$ Memory. arXiv:2112.05682 [cs.LG]

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv:2204.06125 [cs.CV] https://arxiv.org/abs/2204.06125

Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. 2002. Photographic tone reproduction for digital images. *ACM Trans. Graph.* 21, 3 (July 2002), 267–276. https://doi.org/10.1145/566654.566575

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. TEXTure: Text-Guided Texturing of 3D Shapes. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*. Article 54, 11 pages.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. 2023. MVDream: Multi-view Diffusion for 3D Generation. *arXiv:2308.16512* (2023).

Stability AI. 2025. Stable Diffusion 3.5 Large. https://huggingface.co/stabilityai/stable-diffusion-3.5-large. Accessed: 2025-01-15.

Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. 2023. MVDiffusion: Enabling Holistic Multi-view Image Generation with Correspondence-Aware Diffusion. arXiv:2307.01097 [cs.CV] https://arxiv.org/abs/2307.01097

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. 2023. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1921–1930.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Giuseppe Vecchio and Valentin Deschaintre. 2024. MatSynth: A Modern PBR Materials Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 22109–22118.

Giuseppe Vecchio, Rosalie Martin, Arthur Roullier, Adrien Kaiser, Romain Rouffet, Valentin Deschaintre, and Tamy Boubekeur. 2024. ControlMat: A Controlled Generative Approach to Material Capture. *ACM Trans. Graph.* 43, 5, Article 164 (Sept. 2024), 17 pages. https://doi.org/10.1145/3688830

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021), 108:1–108:14. https://doi.org/10.1145/3450626.3459804

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. 2024. SV3D: Novel Multi-view Synthesis and 3D Generation from a Single Image using Latent Video Diffusion. *arXiv:2403.12008* (2024).

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. 2023. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. *arXiv:2305.16213* (2023).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771 [cs.CL] https://arxiv.org/abs/1910.03771

Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T. Barron, and Aleksander Holynski. 2024. CAT4D: Create Anything in 4D with Multi-View Video Diffusion Models. *arXiv:2411.18613* (2024).

Xudong Xu, Zhaoyang Lyu, Xingang Pan, and Bo Dai. 2023. MATLABER: Material-Aware Text-to-3D via LAtent BRDF auto-EncodeR. *arXiv preprint arXiv:2308.09278* (2023).

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. 2024. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer.

Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. 2023. GaussianDreamer: Fast Generation from Text to 3D Gaussian Splatting with Point Cloud Priors. *arxiv:2310.08529* (2023).

Kim Youwang, Tae-Hyun Oh, and Gerard Pons-Moll. 2024. Paint-it: Text-to-Texture Synthesis via Deep Convolutional Texture Map Optimization and Physically-Based Rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. 2024. RGB-X: Image decomposition and synthesis using material- and lighting-aware diffusion models. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*. ACM, 1–11. https://doi.org/10.1145/3641519.3657445

Hongkun Zhang, Zherong Pan, Congyi Zhang, Lifeng Zhu, and Xifeng Gao. 2024b. TexPainter: Generative Mesh Texturing with Multi-view Consistency. arXiv:2406.18539 [cs.CV] https://arxiv.org/abs/2406.18539

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. arXiv:2302.05543 [cs.CV] https://arxiv.org/abs/2302.05543

Shangzhan Zhang, Sida Peng, Tao Xu, Yuanbo Yang, Tianrun Chen, Nan Xue, Yujun Shen, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. 2024c. MaPa: Text-driven Photorealistic Material Painting for 3D Shapes. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA, Article 4, 12 pages. https://doi.org/10.1145/3641519.3657504

Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, and Xiaogang Jin. 2024a. DreamMat: High-quality PBR Material Generation with Geometry- and Light-aware Diffusion Models. arXiv:2405.17176 [cs.GR] https://arxiv.org/abs/2405.17176

Junzhe Zhu and Peiye Zhuang. 2023. HiFA: High-fidelity Text-to-3D Generation with Advanced Diffusion Guidance. *arXiv:2305.18766* (2023).

A ADDITIONAL DISCUSSION

A.1 Relation to SDS-based methods

Score distillation sampling (SDS) offers a way to deal with multi-view consistency by performing alternating steps of diffusion and inverse rendering in an iterative loop. In contrast, our technique provides view consistency while running the diffusion denoiser only once, and inverse rendering occurs after the generation is complete. This makes reconstruction faster than iterative SDS techniques. Also, SDS backpropagates multiple diffused solutions to the 3D representation, whereas our method reconstructs materials from on one sharp target image.

Our contributions for view-consistency (attention biasing, view-correlated noise) are orthogonal to SDS and some applications may benefit from a combined approach. We leave this for future work.

A.2 Choice of viewpoints

All experiments in the paper use roughly equidistant views spaced evenly around the object.

In cases where none of the views observe a particular part of the object, the input material in the unseen area remains unchanged. This is a limitation that we share with all prior works that require rendering 2D images of a 3D scene.

None of the ControlNets we use make assumptions on the positioning of the views. Our simple view placement could be replaced by an adaptive strategy, e.g., to observe otherwise occluded regions or zoom in on specific parts. In the latter case, the size of the pixel neighborhood during attention biasing (Section 4.3 of the main paper) should be adjusted for views that have highly non-uniform area sampling. E.g., in case the same surface region is observed from different views with different distances or zoom levels.

A.3 Memory bottlenecks

Section 5 in the main paper discusses a memory-efficient implementation of our attention biasing approach. Another memory bottleneck arises during the main diffusion process, where high-resolution multi-view grids are passed through the variational autoencoder used for mapping between image and (spatially-downsampled) latent space. For this conversion we found it necessary to split the grid back into tiles consisting of the individual views. That said, the latent sampling inside the autoencoder and the main diffusion and denoising processes can then internally operate on the full resolution (re)concatenated grids.

A.4 Inverse rendering details

We use Mitsuba's differentiable path tracer built on Path Replay Backpropagation [Vicini et al. 2021]. Both the forward and backwards passes are set to 128 SPP. We allow up to 3 bounces of light.

The inverse rendering inputs are the generated diffusion images (used as the target images), and the initial scene parameters (with the initial material textures and fixed geometry and lighting).

The optimized material parameters are albedo, roughness, and normal textures with respective Adam learning rates 0.001, 0.001, and 0.0001 in most cases. The latter value is smaller to avoid strongly tilted normals that otherwise produce black artifacts in the renderings at grazing angles.

A.5 Robustness of inverse rendering

We mitigate the ill-posed nature of lighting and material decomposition by conditioning the diffusion on renderings with the initial material and known illumination. We observe that the generated images preserve the lighting of the conditioning rendering. With known geometry and lighting we are left with updating only the input material textures. This proved sufficient in most of our experiments, but a failure case is shown in Section 7 of the main paper.

B ADDITIONAL EXPERIMENTS

B.1 Comparisons

Figures 11–15 show additional comparisons to related work, similar to Figure 5 in the main paper. Figure 16 shows the corresponding decomposition into the individual material texture maps that each method recovers. Overall, our results are more detailed and more realistic. Please zoom-in for a better view of the details.

RGB↔X is not designed to produce multi-view consistent images, but the editing also leads to unrealistic appearance (Kettle) or results that do not adhere to the prompt (Statue, David Bust).

In case of TexPainter, the multi-view generated images are inconsistent from view to view (see the side of Briefcase, the back of David Bust, the body of Kettle, bottom front of Statue). This leads to blurry reconstructed textures in Briefcase and Kettle, and superimposed effects in Statue.

DreamMat and Paint-it both use variants of SDS, which tend to generate blurry textures due to the inconsistencies of the diffusion model solutions from iteration to iteration. Instead, our approach provides improved multi-view consistency during generation and performs inverse rendering on one sharp target image.

B.2 Visual results

Figures 17–19 presents additional results of our complete pipeline starting from basic assets all the way to the recovered material parameters, as in Figure 10 in the main paper.

Figure 20 shows an extended version of Figure 3 in the main paper, and visualizes exemplary attention scores before and after adding the bias for a single specific surface point.

Figures 21 and 22 are extended versions of Figure 6 from the main paper and demonstrate the full set of user-controlled parameters and their impact on the generated visuals. The *classifier-free guidance* parameter controls the amount of adherence to the prompt; the *ControlNet tile scale* and the added noise parameters control how much the generated material is allowed to deviate from the input views, and trades off the amount of detail vs preservation of the original design intent.

B.3 Warped images

For the results that demonstrate multi-view consistency (Figures 8 and 9 in the main paper), we provide another version with warped images (Figures 23 and 24, respectively). Here, images from one viewpoint are warped to match the other viewpoint for easier visual comparison. This is possible using the known camera parameters and geometry.

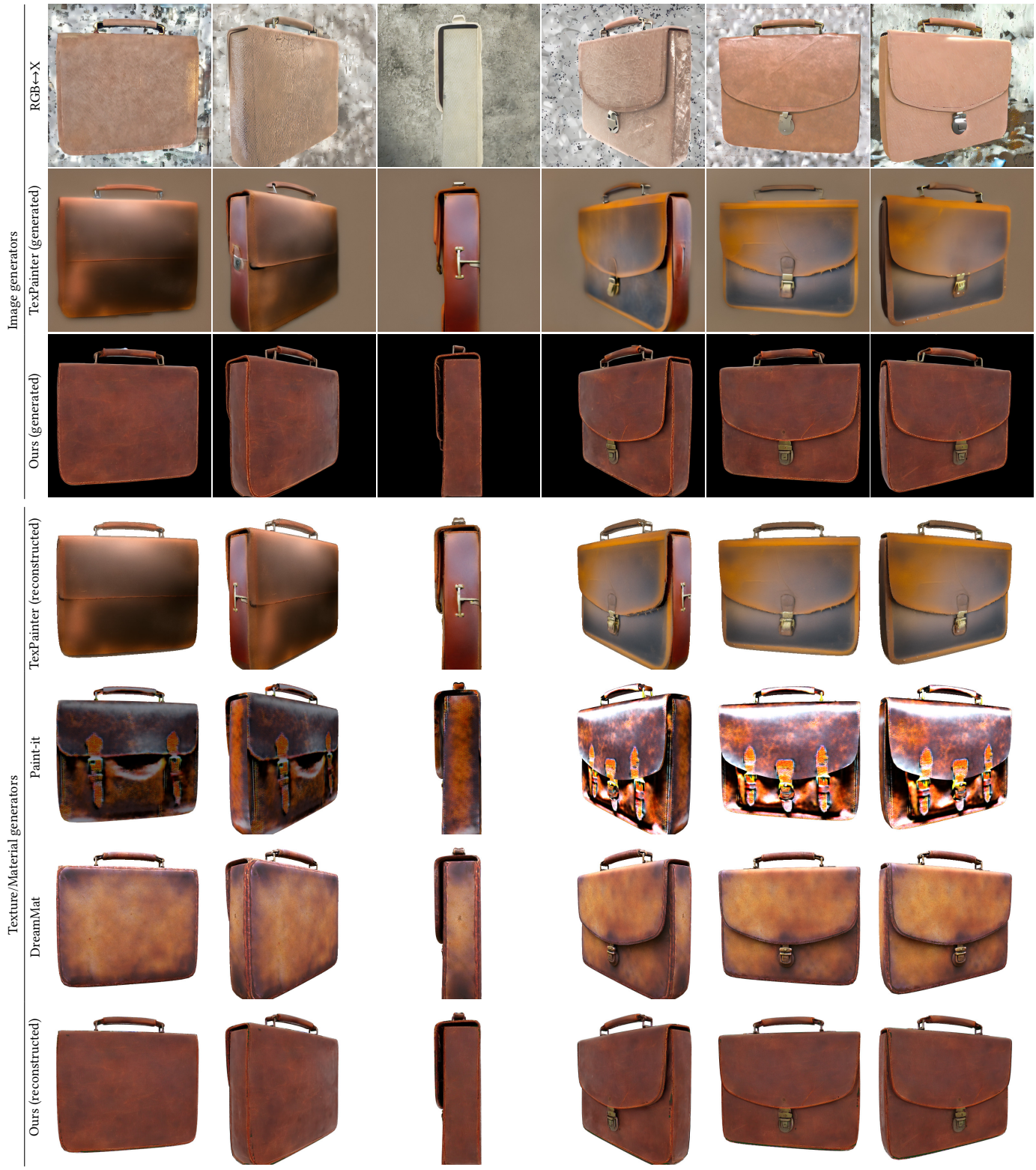


Fig. 11. Prior work comparison on Briefcase. All approaches share the same prompt (as reported in the main results) and the same input mesh.

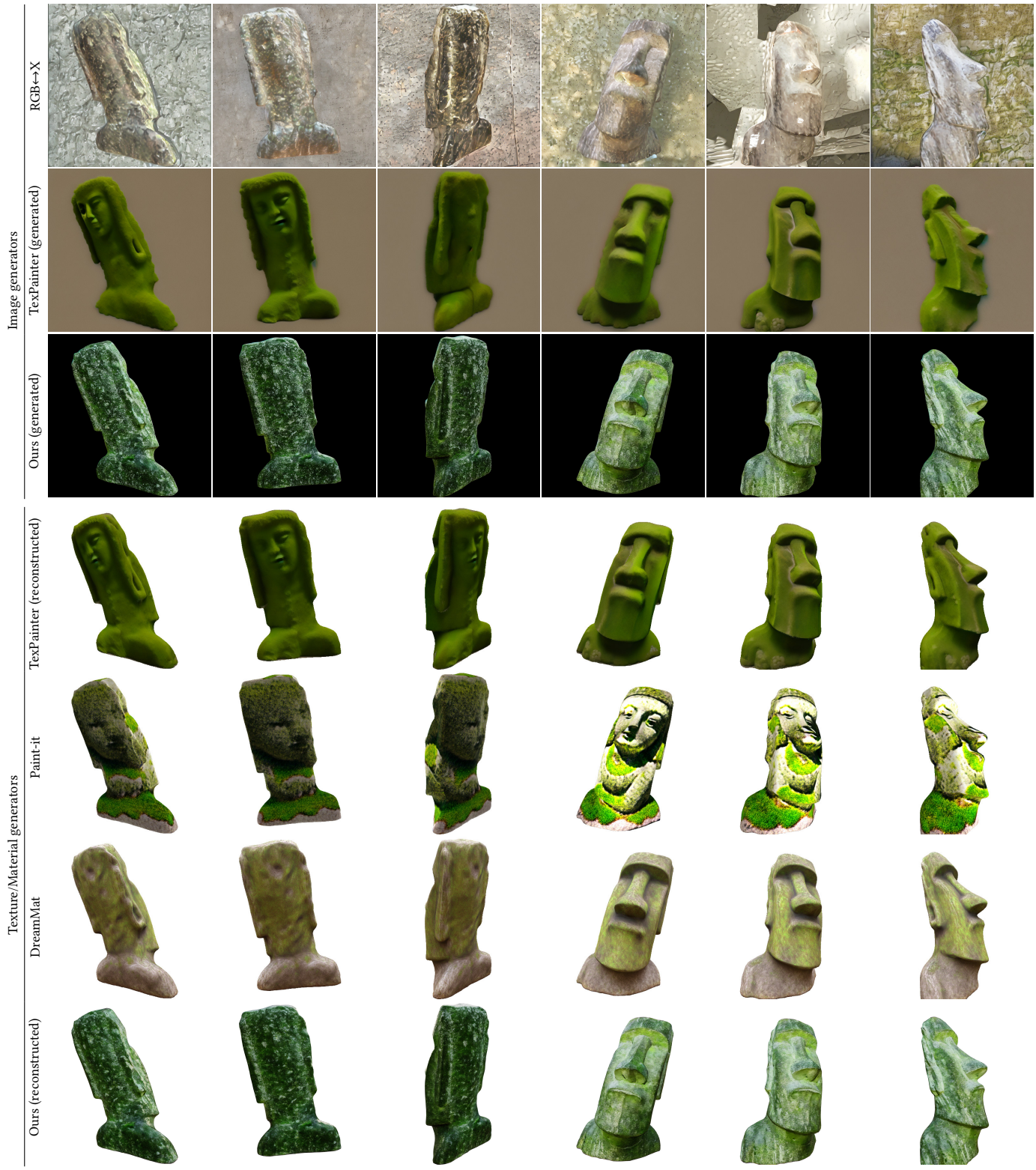


Fig. 12. Prior work comparison on Statue. All approaches share the same prompt (as reported in the main results) and the same input mesh.

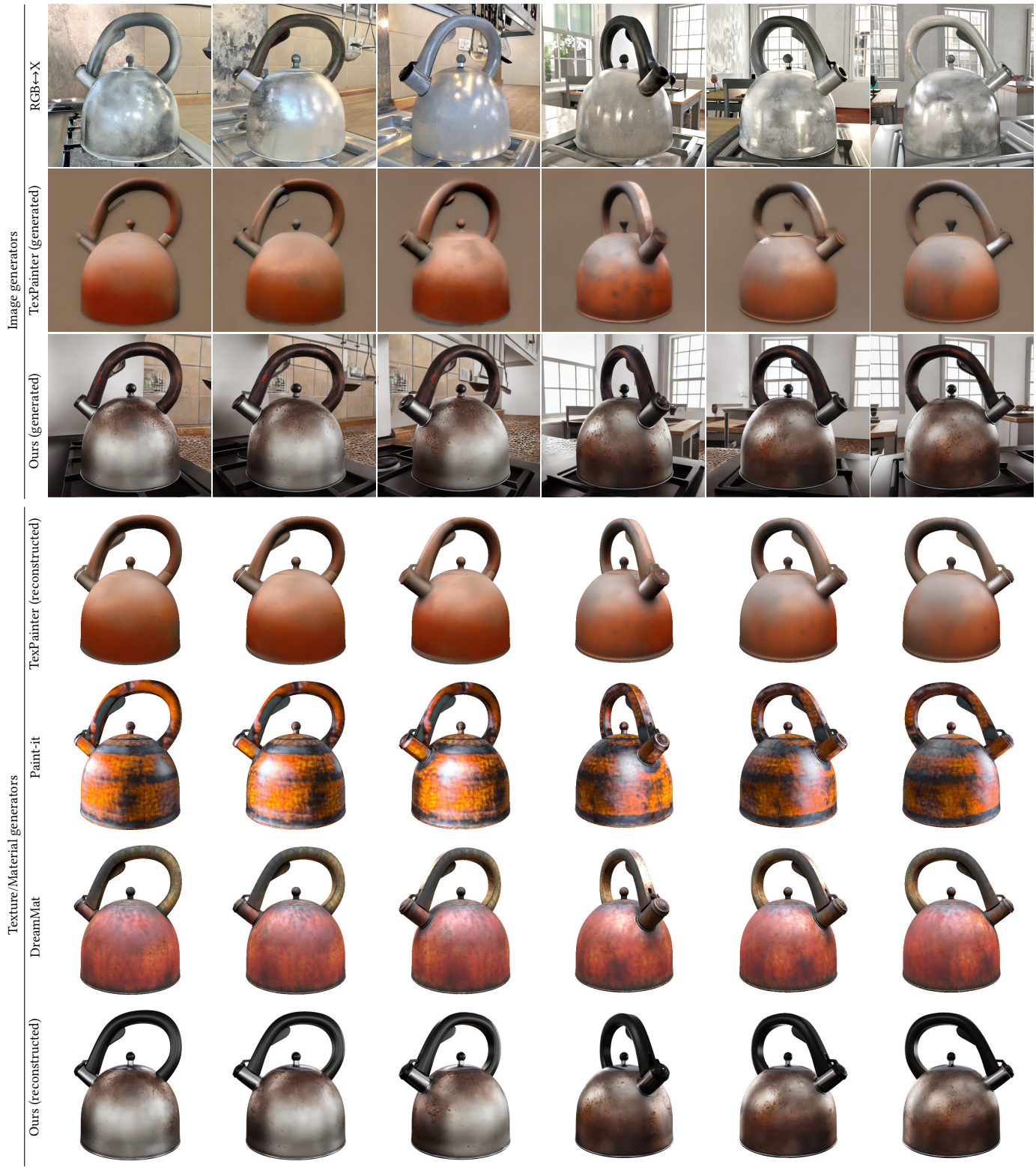


Fig. 13. Prior work comparison on Kettle. All approaches share the same prompt (as reported in the main results) and the same input mesh.

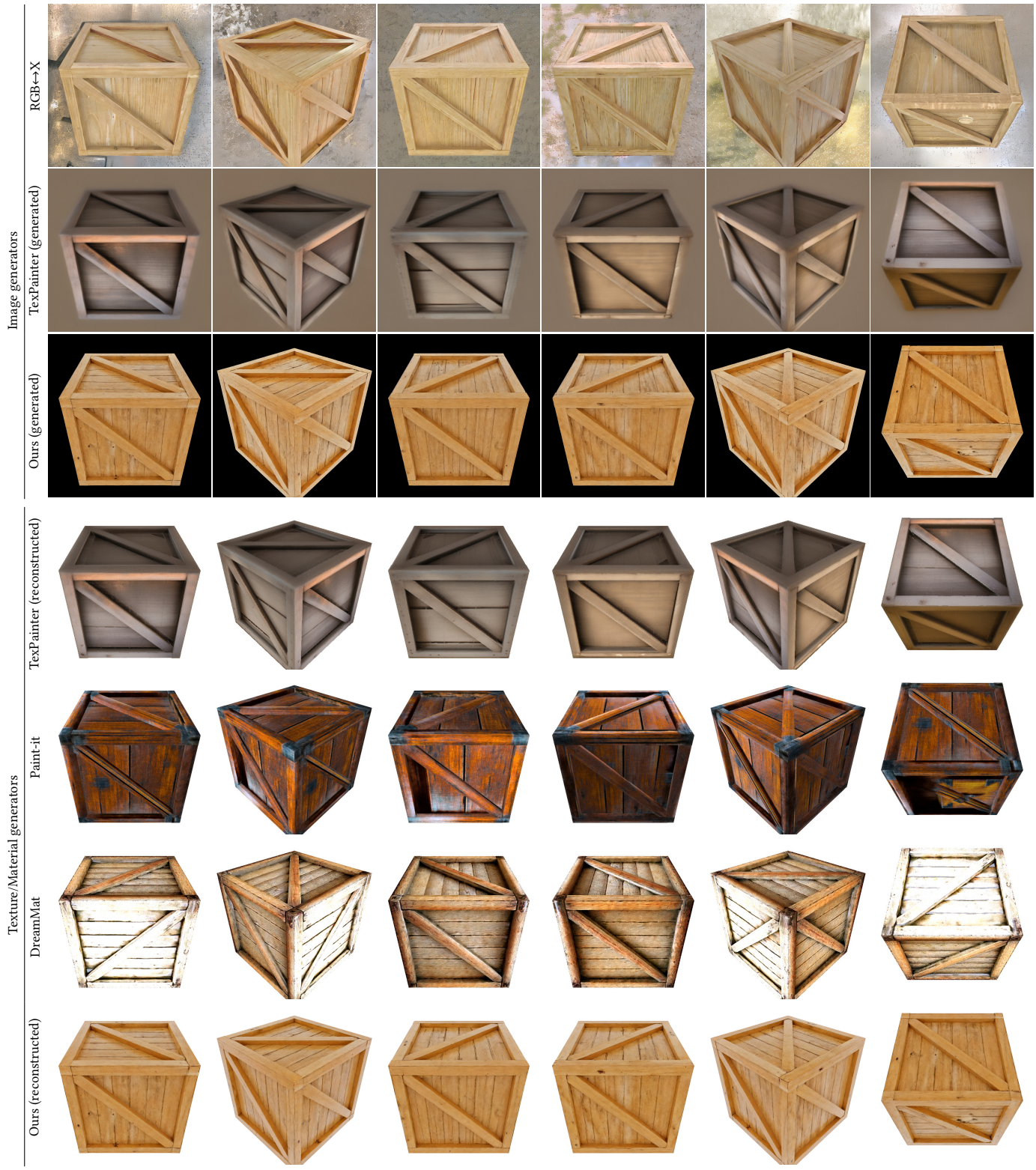


Fig. 14. Prior work comparison on Wooden Box. All approaches share the same prompt (as reported in the main results) and the same input mesh.

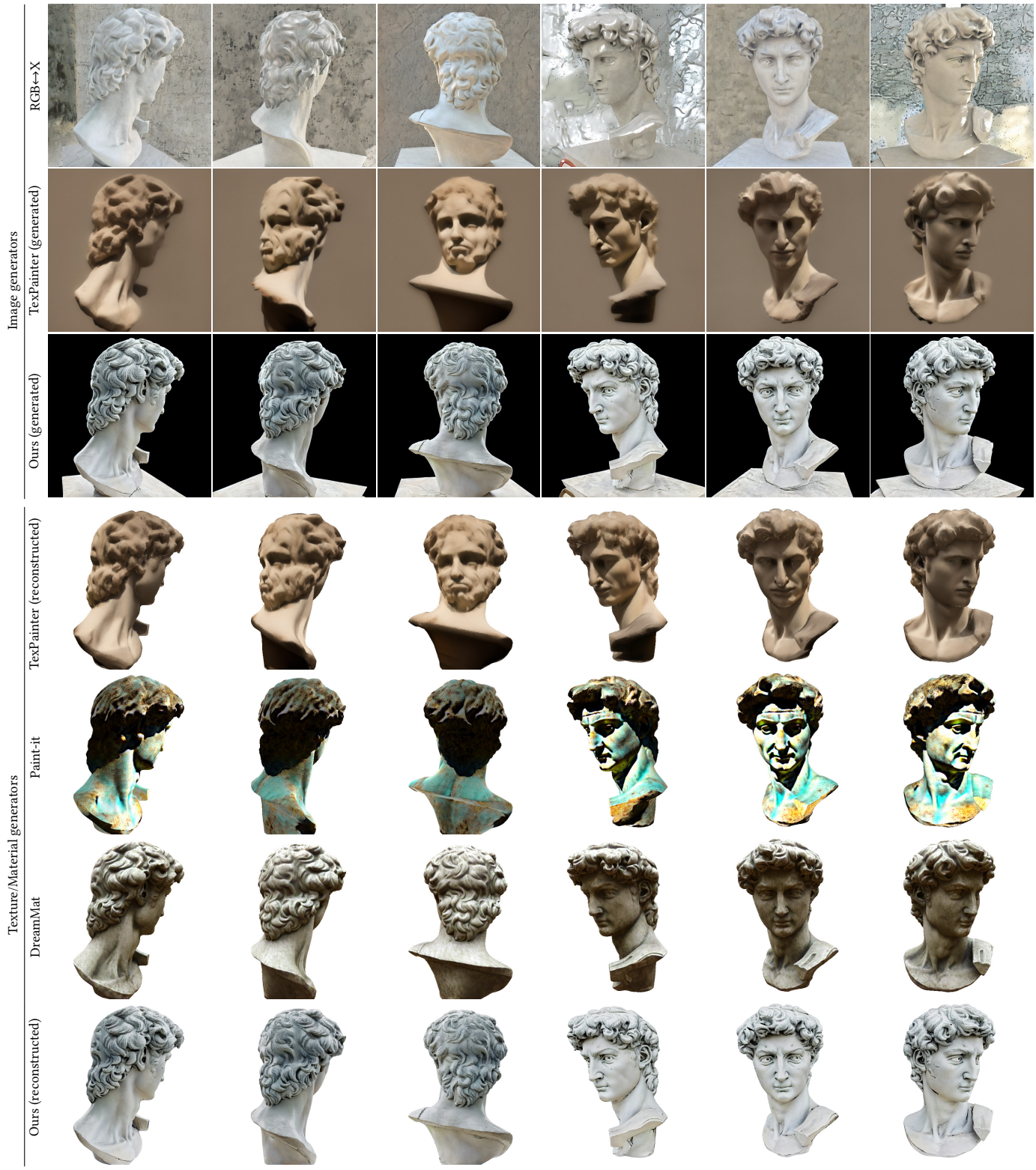


Fig. 15. Prior work comparison on David Bust. All approaches share the same prompt (as reported in the main results) and the same input mesh.

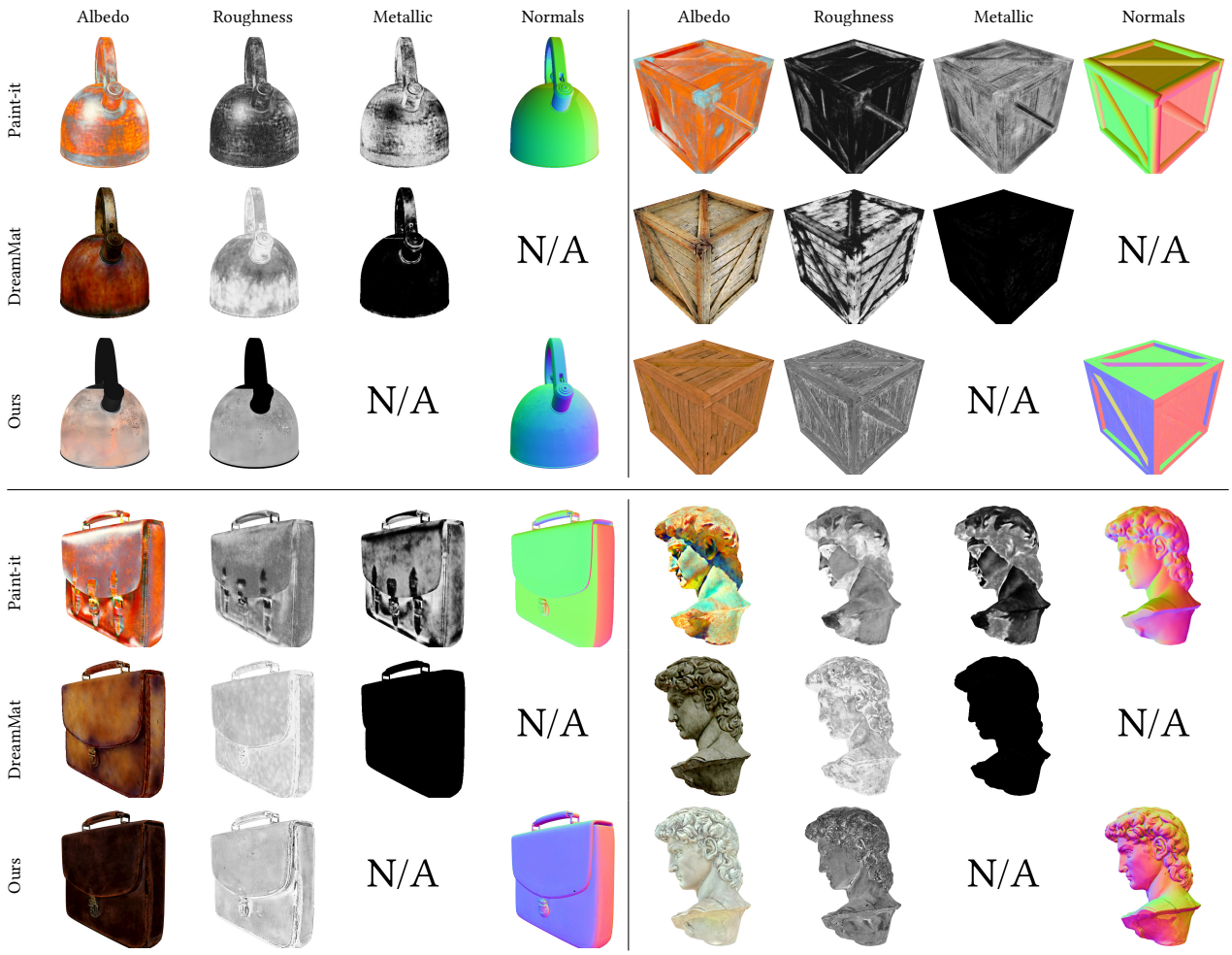


Fig. 16. Material decomposition results of prior works and our method. For prior work we used their default settings; each method uses its own renderer with different inverse rendering parameters, coordinate systems, and material conventions.

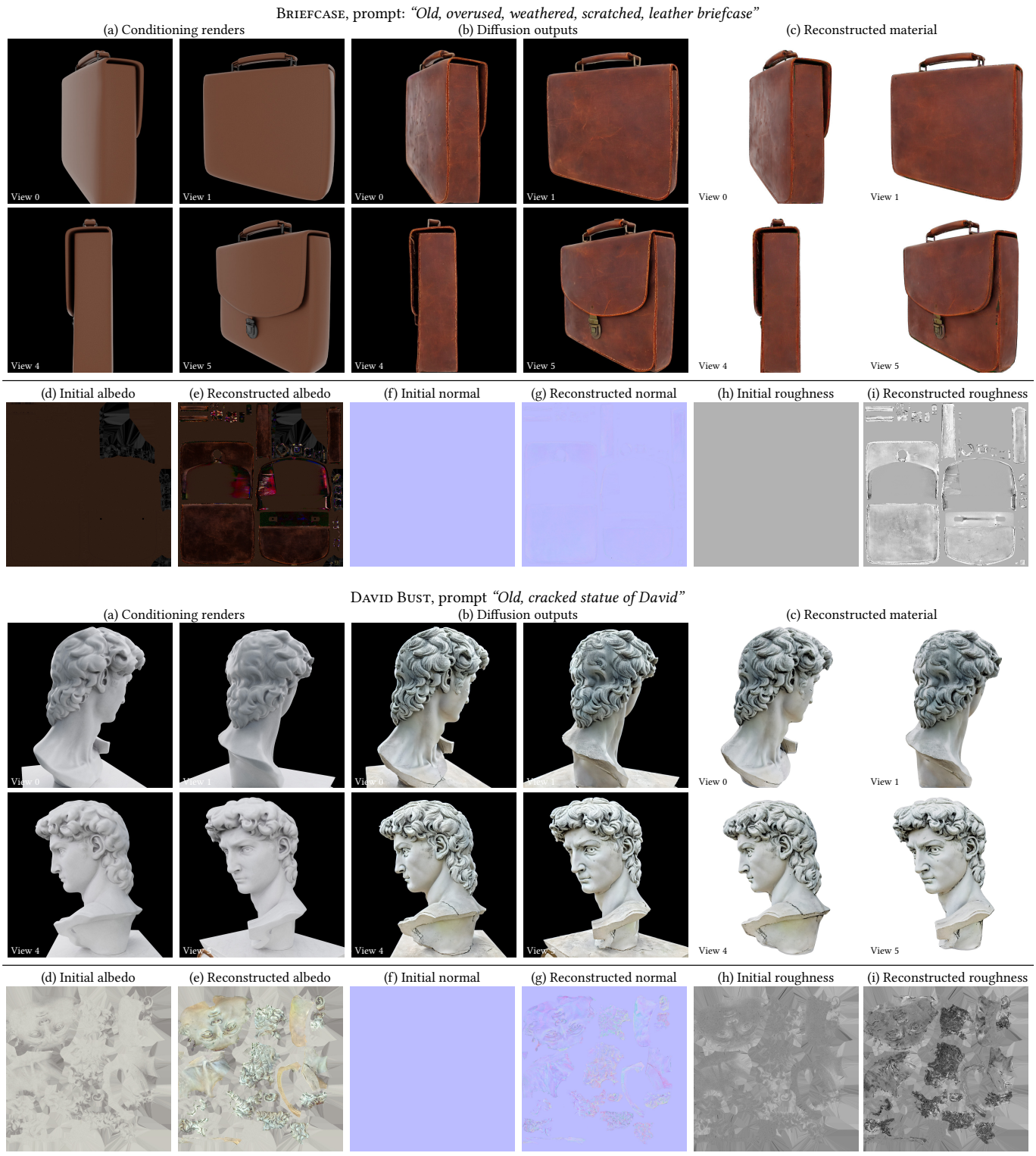


Fig. 17. We use renderings of the original asset (two pairs of two adjacent views are shown in (a)) to condition the diffusion model to produce images with enhanced appearance (b), which is then backpropagated into the original material definition. In (c), we show the resulting asset on four frames from a turntable animation (we picked frames that correspond to the conditioning views); see the accompanying video for the full animation. Columns (d) to (i) show albedo, normal, and roughness textures of the initial and reconstructed material.

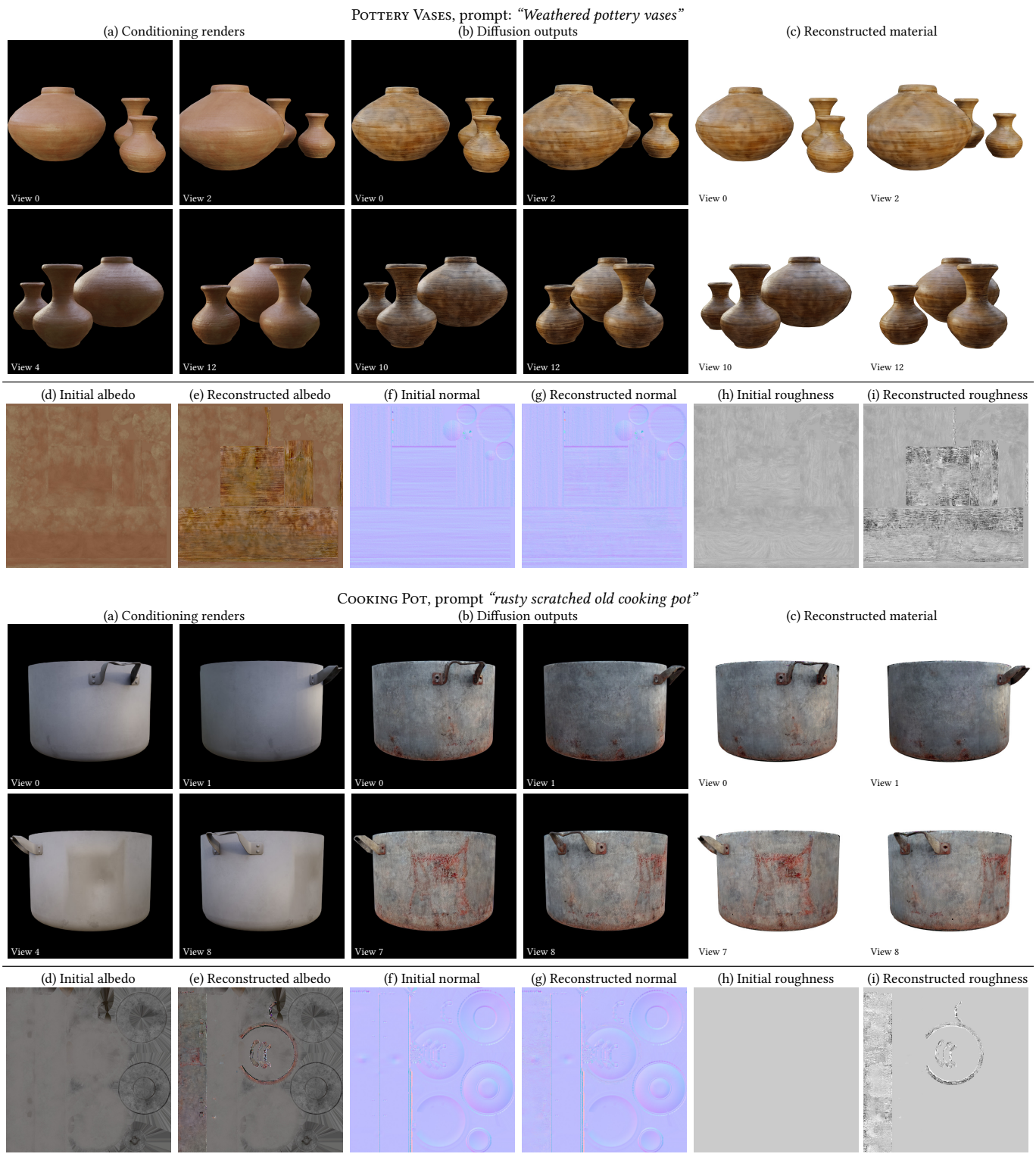


Fig. 18. We use renderings of the original asset (two pairs of two adjacent views are shown in (a)) to condition the diffusion model to produce images with enhanced appearance (b), which is then backpropagated into the original material definition. In (c), we show the resulting asset on four frames from a turntable animation (we picked frames that correspond to the conditioning views); see the accompanying video for the full animation. Columns (d) to (i) show albedo, normal, and roughness textures of the initial and reconstructed material.

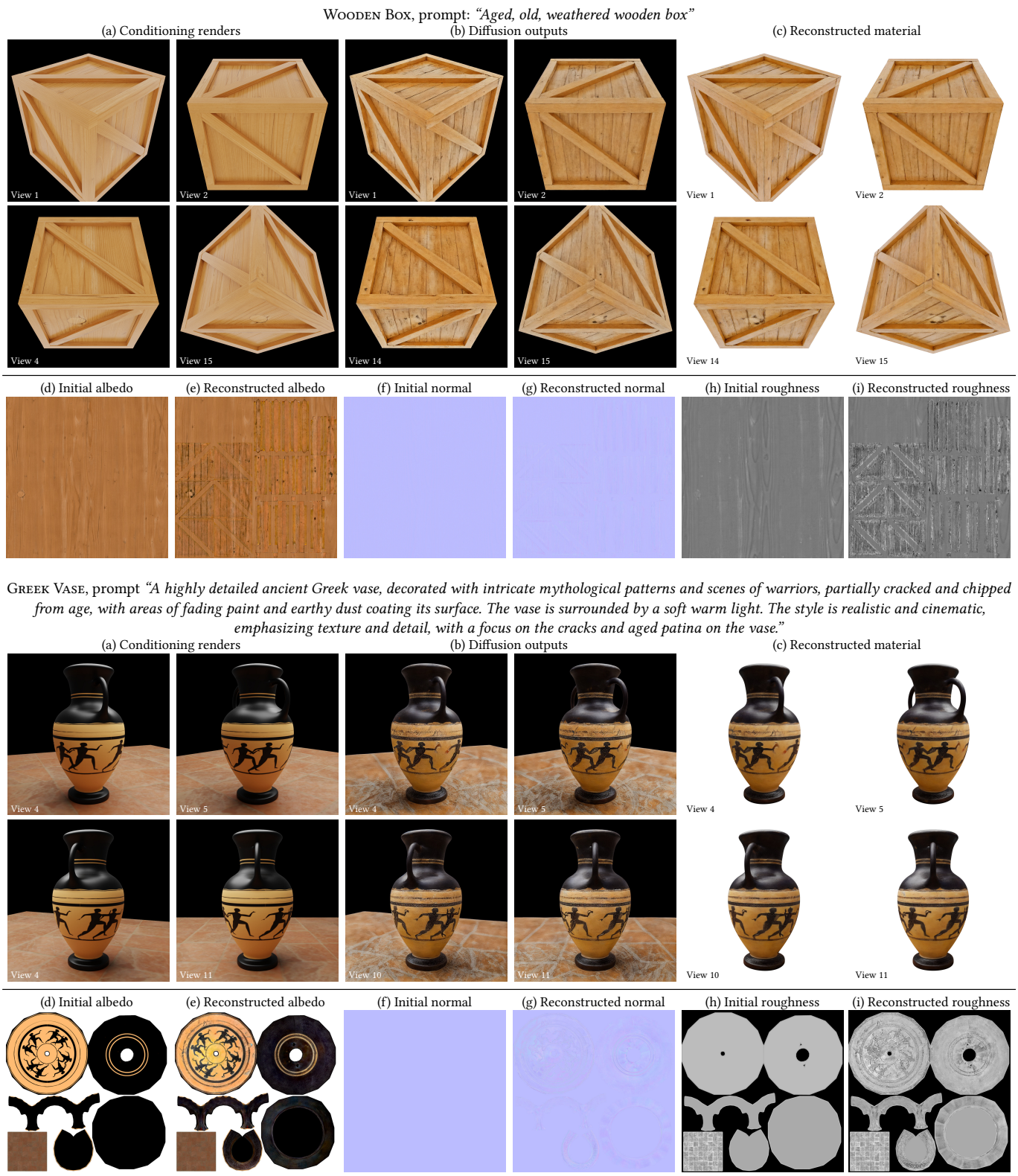


Fig. 19. We use renderings of the original asset (two pairs of two adjacent views are shown in (a)) to condition the diffusion model to produce images with enhanced appearance (b), which is then backpropagated into the original material definition. In (c), we show the resulting asset on four frames from a turntable animation (we picked frames that correspond to the conditioning views); see the accompanying video for the full animation. Columns (d) to (i) show albedo, normal, and roughness textures of the initial and reconstructed material.

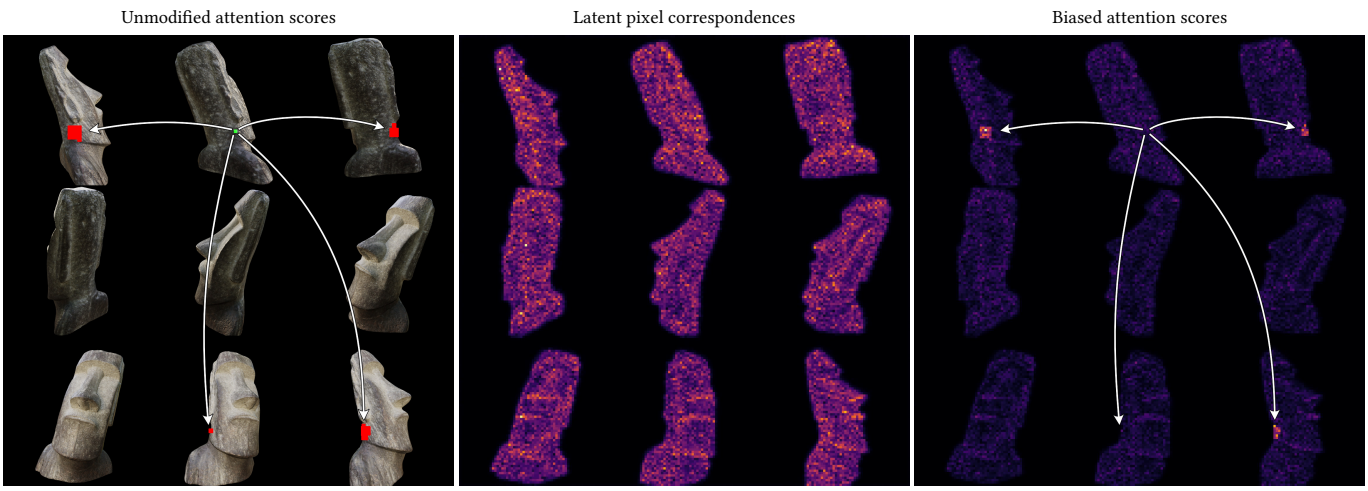


Fig. 20. An extended version of Figure 3 from the main paper. Left: An example latent pixel (green) attends to corresponding image regions in other views (red). Middle: One *row* of the attention score matrix related to that green pixel is rearranged into a false-color image showing how much it attends to all other pixels in one stage of the diffusion model. Right: We bias the matrix elements in *columns* that correspond to the identified red regions to promote attention—and hence consistency—between these latents. Scores are visualized after the softmax in Eq. (1) and (2) and gamma-mapped for clarity.

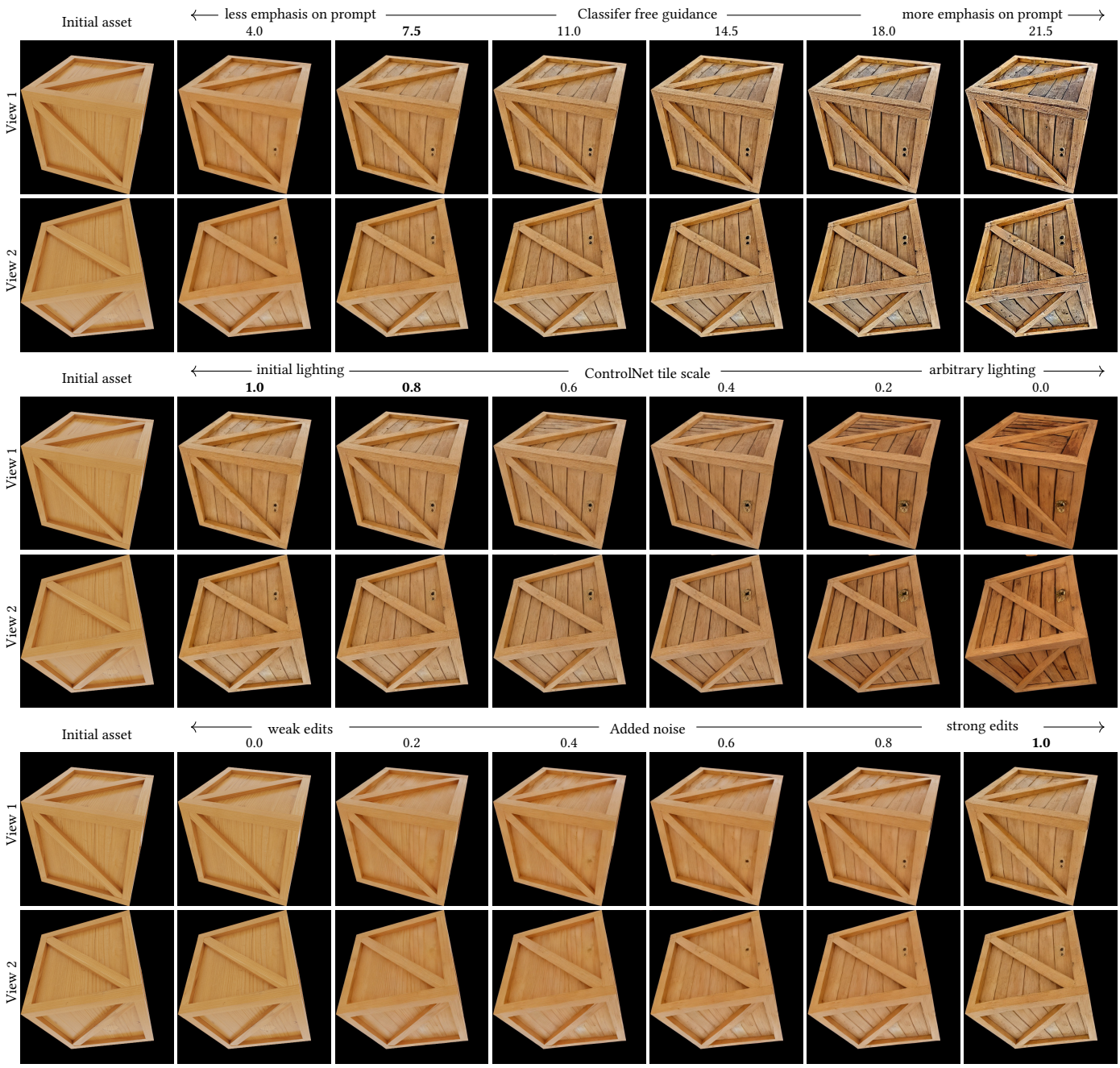


Fig. 21. Expanded version of Figure 6 from the main paper on Wooden Box. Apart from the *classifier-free guidance* parameter (top), the underlying diffusion models of our system expose two further parameters that can be tweaked by users. Both the strength at which the *ControlNet tile* (middle) and the *input noise* (bottom) are applied allow tweaking how much the generated material details will deviate from the base material. Numbers highlighted in bold indicate parameter ranges we use throughout the results in the paper.

Fig. 22. Another example of Figure 21 on Statue. Apart from the *classifier-free guidance* parameter (top), the underlying diffusion models of our system expose two further parameters that can be tweaked by users. Both the strength at which the *ControlNet tile* (middle) and the *input noise* (bottom) are applied allow tweaking how much the generated material details will deviate from the base material. Numbers highlighted in bold indicate parameter ranges we use throughout the results in the paper.

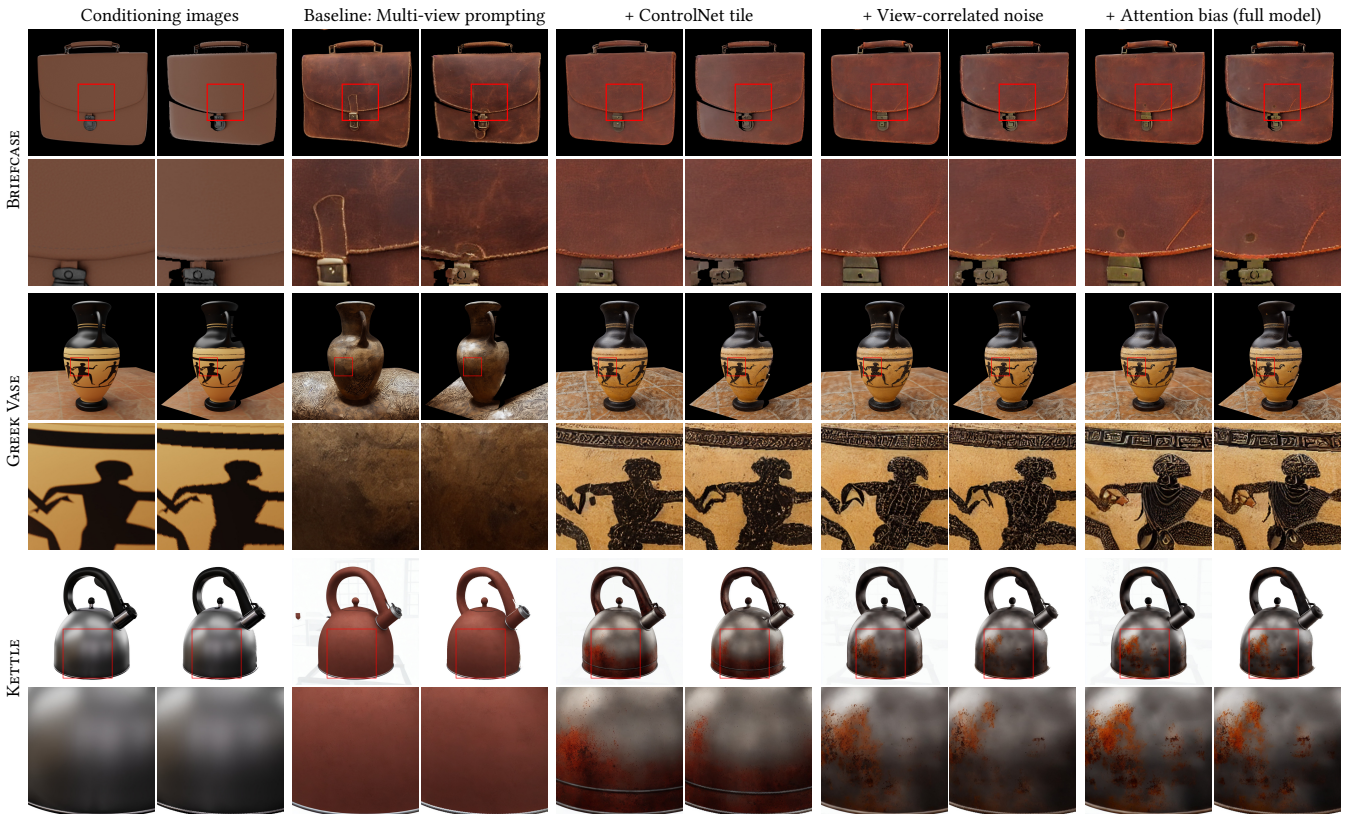


Fig. 23. Another version of Figure 8 from the main paper. Here, images of the right view are warped to match the viewpoint in the left view for easier visual comparison. We also added the Kettle scene.

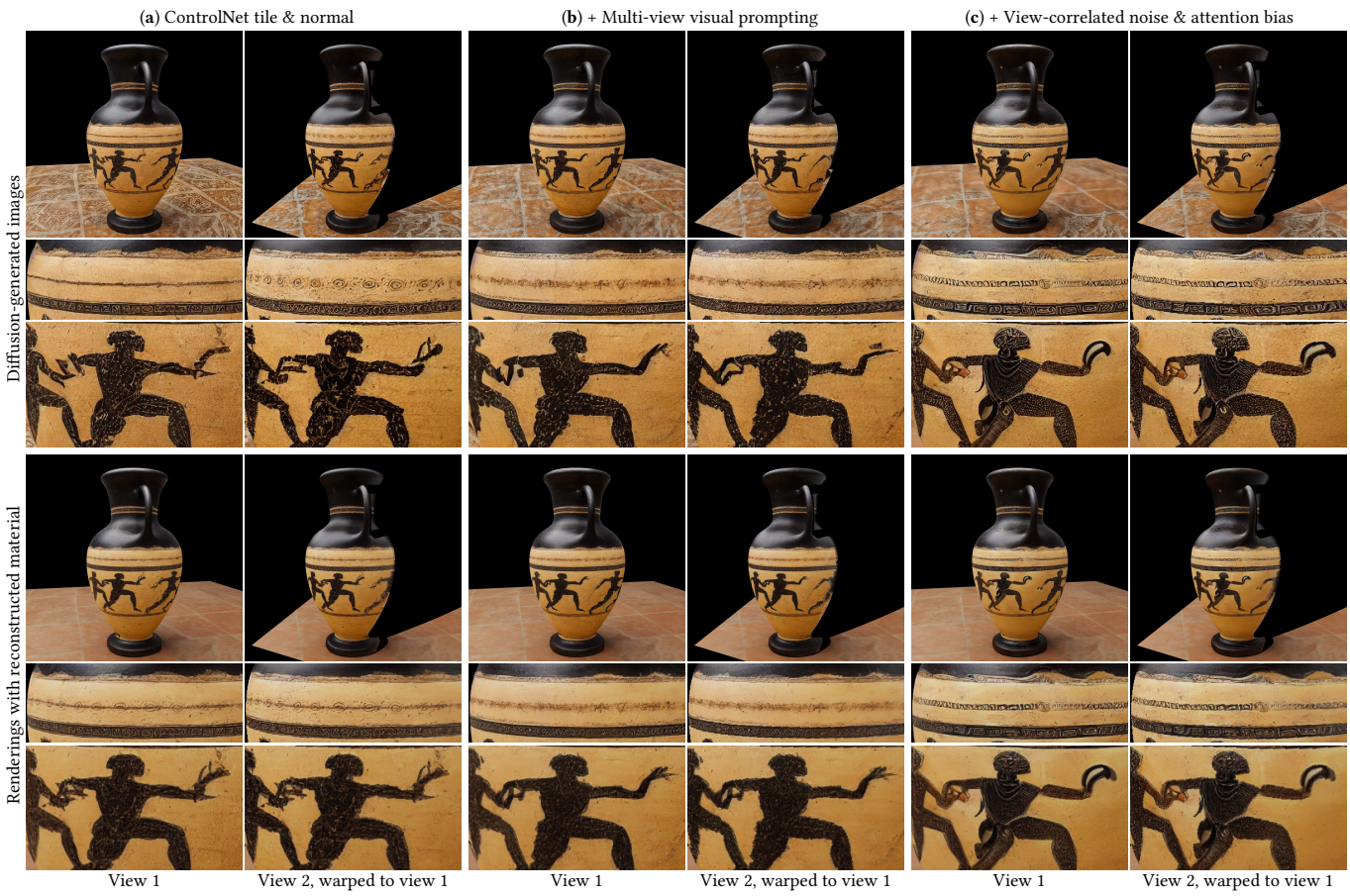


Fig. 24. Another version of Figure 9 from the main paper. Here, images of view 2 warped to match the viewpoint of view 1 for easier visual comparison.