Dequantization and Color Transfer with Diffusion Models

Vaibhav Vavilala
UIUC
vv16@illinois.edu

Faaris Shaik
UIUC
faariss2@illinois.edu

David Forsyth
UIUC
daf@illinois.edu

Abstract

We demonstrate an image dequantizing diffusion model that enables novel edits on natural images. We propose operating on quantized images because they offer easy abstraction for patch-based edits and palette transfer. In particular, we show that color palettes can make the output of the diffusion model easier to control and interpret. We first establish that existing image restoration methods are not sufficient, such as JPEG noise reduction models. We then demonstrate that our model can generate natural images that respect the color palette the user asked for. For palette transfer, we propose a method based on weighted bipartite matching. We then show that our model generates plausible images even after extreme palette transfers, respecting user query. Our method can optionally condition on the source texture in part or all of the image. In doing so, we overcome a common problem in existing image colorization methods that are unable to produce colors with a different luminance than the input. We evaluate several possibilities for texture conditioning and their trade-offs, including luminance, image gradients, and thresholded gradients, the latter of which performed best in maintaining texture and color control simultaneously. Our method can be usefully extended to another practical edit: recoloring patches of an image while respecting the source texture. Our procedure is supported by several qualitative and quantitative evaluations.

1. Introduction

We describe image editing procedures for modifying the color of natural images. Our editing procedures allow artists to apply natural vector art edits to pixel art. For example, an artist can modify the color palette of an image - including the colors in the palette and which patches are assigned to each color. Spatial layout control of reference image based color transfer is not a common capability of existing methods. We use diffusion models to "snap" the artist's changes to realistic images.

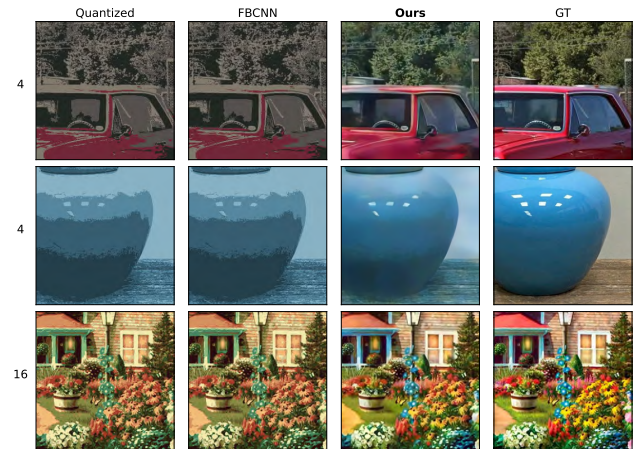Diffusion models [5] have proven to be SOTA in image

Figure 1. We present an image dequantizer that outperforms existing image restoration models. A quantized image is shown in the first column; the second column shows the results of denoising by [15]; the third is from our procedure (which only requires the quantized image and number of palette colors as input - 4, 16, 64 here); the fourth shows GT. Our model is willing to deviate from the input to produce smoother gradients and life-like colors. Dequantization is not commonly investigated in the image restoration community - here we show why specialized dequantization models are useful, and that existing methods are not sufficient.

synthesis. In particular, techniques have been developed to exert control over the output via text prompts [28], edges, segmentation maps, or depth maps [45]. Some of these procedures do not apply here – for example, there is no method for editing an image's depth map, and important and natural controls are not well-explored.

Fine-grained edits to an image involve selecting some portion of the image, changing it, then obtaining a natural result. Diffusion models have been shown to accept segmentations as conditioning ([27, 45]), but fine-grained edits have not yet been shown. Our method accepts a segmentation from any source, such as [21]. A user can then remove one or more patches, and the diffusion model will inpaint the result. Further, the user can specify a color to be used for patches, and the diffusion model will respect that

control. This color conditioning is not a standard control in image inpainting (as opposed to text prompts, which are well-established). We argue that changing the color of objects in a scene while maintaining texture is useful in artist workflows. We show that our color & texture -conditioned inpainting method works well in practice.

Palette control involves applying a specified palette to a given image. The palette may come from a pre-built colormap, or more commonly, an example image. Palette control may involve very large changes to the color palette, changing the overall feel of the image (for example, Fig. 8) without disrupting gloss effects, color gradients, and so on.

Current style transfer methods (reviewed below) do not apply, because they do not offer spatial layout control over which image patch gets what color. Further, our method offers varying levels of palette abstraction (the number of colors in the palette). Prior methods cannot produce the variety and level of control over the synthesis that our method can.

For a palette control method to be useful in our application, it should be able to make very large changes to the palette of the image; it should be easy to obtain multiple distinct transfers; and it should be possible to apply detailed edits to the palette mapping. Our palette control method is built using three tools: vector quantization of an image's palette; correspondence between color palettes; and "vector dequantization," where a diffusion model constructs one or more natural images conditioned on a vector quantized input. Assume we wish to show image A in B's palette. We vector quantize each image's color gamut. We then build a correspondence between the centers, then map the colors using the correspondence to obtain a vector quantized version of the result. Finally, we vector dequantize to obtain a natural image. Choices that affect the result are: the number of centers chosen in vector quantization; the particular correspondence process; whether or not to condition on texture (and which method if so); and the randomness inherent in the diffusion model.

2. Related Work

Image Synthesis: There is a rich history of image synthesis with GANs [16, 19, 34, 36, 40] and diffusion models [5, 33]. The quality of the generated distributions is often quite good, with problems like aliasing [18] and sparse datasets [17] well-explored. Techniques have been developed to condition the generation on additional inputs like depth, normals [7, 45], and even 3D primitives [37–39].

Image inpainting is a process by which a patch of pixels that has been removed in an image (say for object removal) must be filled in (see [6] for a review). A number of methods have been developed to solve this problem with CNNs [24] and diffusion models [27, 43], generating sensible pixels consistent with the surroundings while preserv-

ing texture outside the target region. Masks can be specified to dictate what region to inpaint, and optional text captions can accompany the diffusion model to guide the synthesis. Inpainting is related but different from colorization, which typically takes as input a grayscale image and optionally color hints via keypoints, and synthesizes *ab* channels to accompany an input *L* channel [23, 46]. In contrast, our method accepts a color-quantized image or a natural image with a patch removed for inpainting, with a color hint encompassing the whole patch (improving spatial control). Of critical importance, our method supports luminance changes, whereas existing methods operating in *Lab* space cannot.

We are not aware of an inpainting method that allows the user to specify the approximate color of the inpainted region as an RGB triple with potentially different luminance (previous work only conditions on a text caption e.g. "red bird," or a limited number of colors and objects at low resolution [20]). We argue this is an important image editing activity and useful to have in practice. In this work, we show this edit can be accomplished with ControlNet, and a segmenter like SegmentAnything [21, 45]; it's likely other inpainting methods can synthesize images conditioned on color, although no one has shown this capability.

Palette mapping can be seen as a variant of style transfer, though we deprecate this view. Style transfer methods (e.g. [9, 10, 13, 22, 26, 31]) do change palettes, but change the texture of the image as well; in contrast we want to change *only* the colors. Some methods require finetuning a style transfer model for each new style [8, 47]; others can generalize to new styles [4, 9–11, 14, 35]. [44] uses text prompts rather than images to guide the style transfer. There is a small literature of palette transfer methods (with a review, [25]). Chang *et al.* demonstrate the value of quantizing a palette in a wholly interactive method. A number of automatic methods are framed as a warp that matches color spaces. Reinhard *et al.* match moments in *Lab* color space [30]; they obtain improved transfer by segmenting image and reference and matching segments. Wu *et al* use an explicit semantic representation to match (so flowers to flowers and grass to grass, say) [42]. Pitié *et al.* warp the color space using repeated 1D projections and a form of histogram matching [29]; Hwang *et al.* use moving least squares [12]. Cho *et al.* train a network to apply a specified palette to a natural image using color augmentations [3]. While palette mapping could be viewed as a colorization problem (decolor, then colorize with the specified palette), we are not aware of colorization algorithms that can be forced to use a restricted palette. Existing style transfer methods typically disrupt texture patterns, as well as change color in uncontrolled fashion. Basioti *et al.* dequantize small face images using a GAN [1].
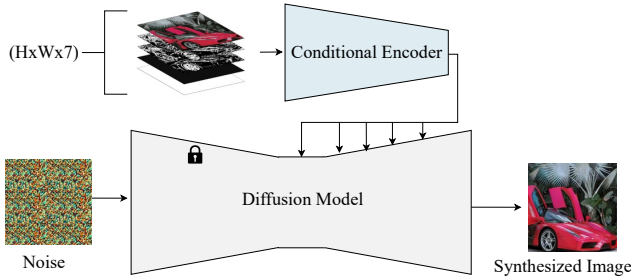
Figure 2. Overview of our method. We build a controlled image synthesis pipeline to dequantize images, optionally with texture conditioning. We can apply palette transfer or inpaint patches with a user-specified color, opening new creative applications.

3. Method

Dataset: We collect 1.8M images from LAION-Aesthetic[1] for training, and hold-out a validation and test set. This dataset spans natural images, capturing a wide range of lighting conditions, subjects, and textures. An array of stylistic artwork is represented, such as digital paintings.

Architecture: Our methodology relies on the principles of ControlNet [45], whereby we introduce additional channels to an existing diffusion model. Core to our method is the concept of scale - pretrained diffusion models have seen billions of images, and we'd like to leverage that knowledge. While image restoration models exist for a variety of degradations like deblurring and noise removal, specialized denoisers have been shown to be useful for specific kinds of noise, such as Monte Carlo noise [41]. Here, we aim to remove quantization noise with diffusion models, and show why that is useful. Our foundation model in this work is DeepFloyd Stage II[2] [33], upon which we build an encoder to accept additional channels as conditioning. As shown in Fig. 2, the base diffusion model is frozen, and the auxiliary encoder module is trainable, extracting useful information from the conditioning and influencing the image synthesis of the base model by adding information to feature maps at varying spatial scales.

Conditioning Image Dequantizer: (1st 3 channels)) The primary function of our diffusion model is to dequantize images. Thus, the input to our conditional encoder is the 3-channel RGB image, that contains the quantized image. We use the median cut algorithm from the PIL library to quantize, selecting the number of colors to be powers of 2 from $N \in \{2^2, 2^3, ..., 2^7\}$. **($4^{th}$ channel)** We wish to tell the network the quantization level since we know it in advance - we add a fourth channel and let every value be a scalar that takes on the value $N/256$. Op-

tionally, we wish to condition on texture, such that a user can preserve source texture when applying a palette transfer or colorization. Conditioning on luminance is the most natural choice, since existing colorization work uses this channel as conditioning. **(5th & 6th channels)** Therefore we train one variant (ours $-$ L) where the fifth channel is the luminance of the GT image at training time. Optionally, at inference time we can replace the luminance of the generated image with the source luminance - we call this configuration ours $-$ L $-$ post and note that it doesn't warrant training a separate model. One can observe that this form of texture conditioning limits the extent of palette transfers (say for example, a user wants to change a white purse to black while preserving texture; luminance conditioning will prevent the user from getting what they asked for). Thus we train another variant - image gradient conditioning, ours $-$ G. We allocate the fifth and sixth channels of the encoder input to be the image gradients of the luminance channel in the x- and y-directions respectively. While this conditioning modestly improved the range of color transfers we could achieve, the sign and magnitude of the image gradient still encodes local luminance information, which in practice limits the success of extreme palette transfers. Consequently, we train a fourth variant, thresholded gradient: ours $-$ T. We simply threshold the absolute value of the image gradients, resulting in binary images in the fifth and sixth channels consumed by our trainable encoder. $L(i,j) > 8$ worked well in practice, where L is the luminance of the input. As our evaluations show, at the cost of a small amount of texture accuracy, we can better decorrelate local luminance, texture, and color with the thresholded gradient approach. To evaluate our dequantization approach, we compare with recent image restoration methods (see Fig. 3), designed to remove JPEG artifacts and general image noise [2, 15]. Evaluations with these works are done in the regime where texture is not supplied to the model, for fair comparison. **(7th channel)** We introduce a seventh channel, a binary texture indicator, to tell the network if texture is available at a particular pixel i, j. Thus we are training three independent networks, $L(\theta; \mathbf{c})$, $G(\theta; \mathbf{c})$, $T(\theta; \mathbf{c})$ with identical settings except the texture conditioning. θ refers to the parameters of the network, including the frozen base diffusion model and trainable encoder; $\mathbf{c}$ refers to the 7 channel auxiliary conditions described above. Diffusion models are also conditioned on the time step $\mathbf{t}$ and text prompt (we use the empty string).

Inference - Palette Transfer: Once we establish that our method can effectively dequantize, then we can use color palettes as an effective abstraction for image editing. For example, we can edit one or more colors in the palette, or which patches get assigned which color. Further, our method can dequantize images over a wide range of colors

---

in the palette, supporting varying levels of abstraction. To perform a palette transfer, we can quantize a source image and extract its palette. From there, we select a target color map, which could be extracted from another image. Then we perform weighted bipartite matching, assigning each source color the most similar target color, ending up with a 1:1 correspondence. Given this modified quantized image, we can then dequantize using our diffusion model to obtain the palette-transferred result. The exact palette matching procedure is up to the artist - our focus in this work is to show that our model respects the palette that was asked for, even extreme palette transfers. We show results for two color-matching algorithms. We call matching based on the most similar color as the `color` method; by negating the pairwise distance matrix between source and target colors, we can effectively match based on *most dissimilar color*, which results in interesting effects (`negative − color`).

Color-conditioned inpainting: Our model naturally lends itself to solve a problem closely related to image dequantization: image inpainting. In particular, we can condition the missing patch on a color of the user's choosing, and optionally preserve source texture when inpainting the patch, which is not an edit discussed in the inpainting literature [32]. To support color-conditioned inpainting, we modify the training process of the existing three networks as follows. We supply the GT image for the first 3 channels, but randomly replace a patch of the image with the mean color of its support. We set the color indicator channel to 1 where GT color is available, and $1/256$ for the masked out patch. Similarly, we can randomly set the texture channels to 0 within the mask. Thus each foundation model that we trained, $\mathsf{L}, \mathsf{G}, \mathsf{T}$, is capable of dequantizing a wide range of images and colors in the palette as well as performing color-conditioned inpainting - including respecting or ignoring source texture based on user request. As we show in our experiments, even though the network is supplied with the mean color within the patch during training, at inference time any color can be supplied, even if it would appear strange in a natural image, and our method respects what was asked for.

Training Details: We train our three models for 2 epochs each on a single A40 GPU, requiring approx. five days. For both inpainting and dequantization, the network is trained to predict the GT image given the conditioning, using standard losses from [33]. We use AdamW optimzer, mixed-precision training, batch size 12, and learning rate 0.00001. Despite the limited amount of training, we achieve highly competitive and useful results. Even though our model was trained on 256-res crops, it supports any aspect ratio at inference time. Inference at 256-res requires approx. 3 seconds on an A40 GPU; inference at $1920{\times}1080$ requires approx. 63 seconds. We found that the $super27$ DDPM sampler performs well, and we experimentally set
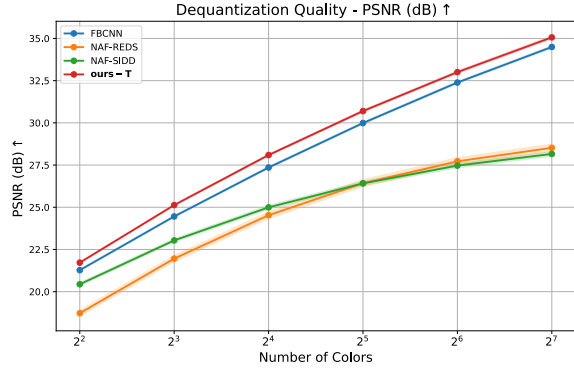

Figure 3. Dequantization without texture conditioning. Our procedure outperforms existing image restoration models across a wide range of palette sizes.

the dynamic thresholding p and c hyperparameters to 0.95 and 1.0 respectively. We remark that our early experimentation showed that our approach works just as well on latent diffusion models like Stable Diffusion [7].
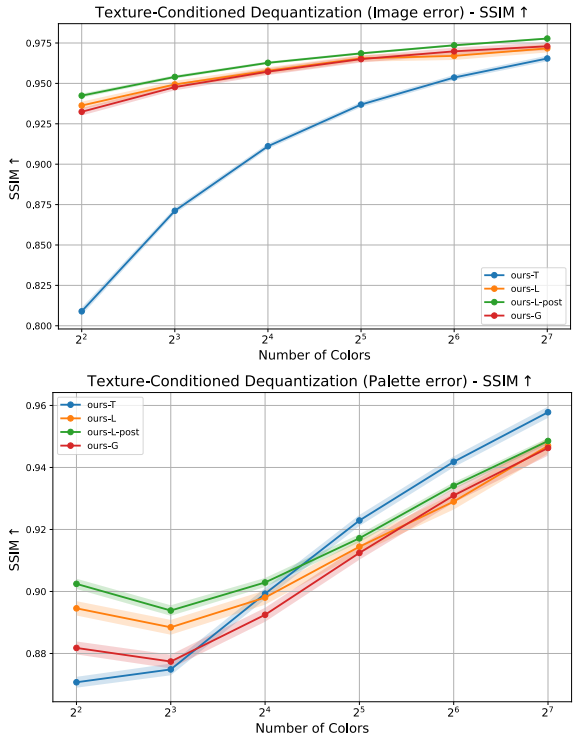

Figure 4. Dequantization with texture conditioning. **Top** All methods can dequantize fairly well, though $\mathsf{ours} - \mathsf{T}$ suffers in reconstruction quality. **Bottom** When measuring error w.r.t. to quantized output vs. quantized input, all methods generally excel, though the fact that $\mathsf{ours} - \mathsf{T}$ holds the least information and performs better in some cases indicates imperfections in the quantization algorithm (we use median-cut).
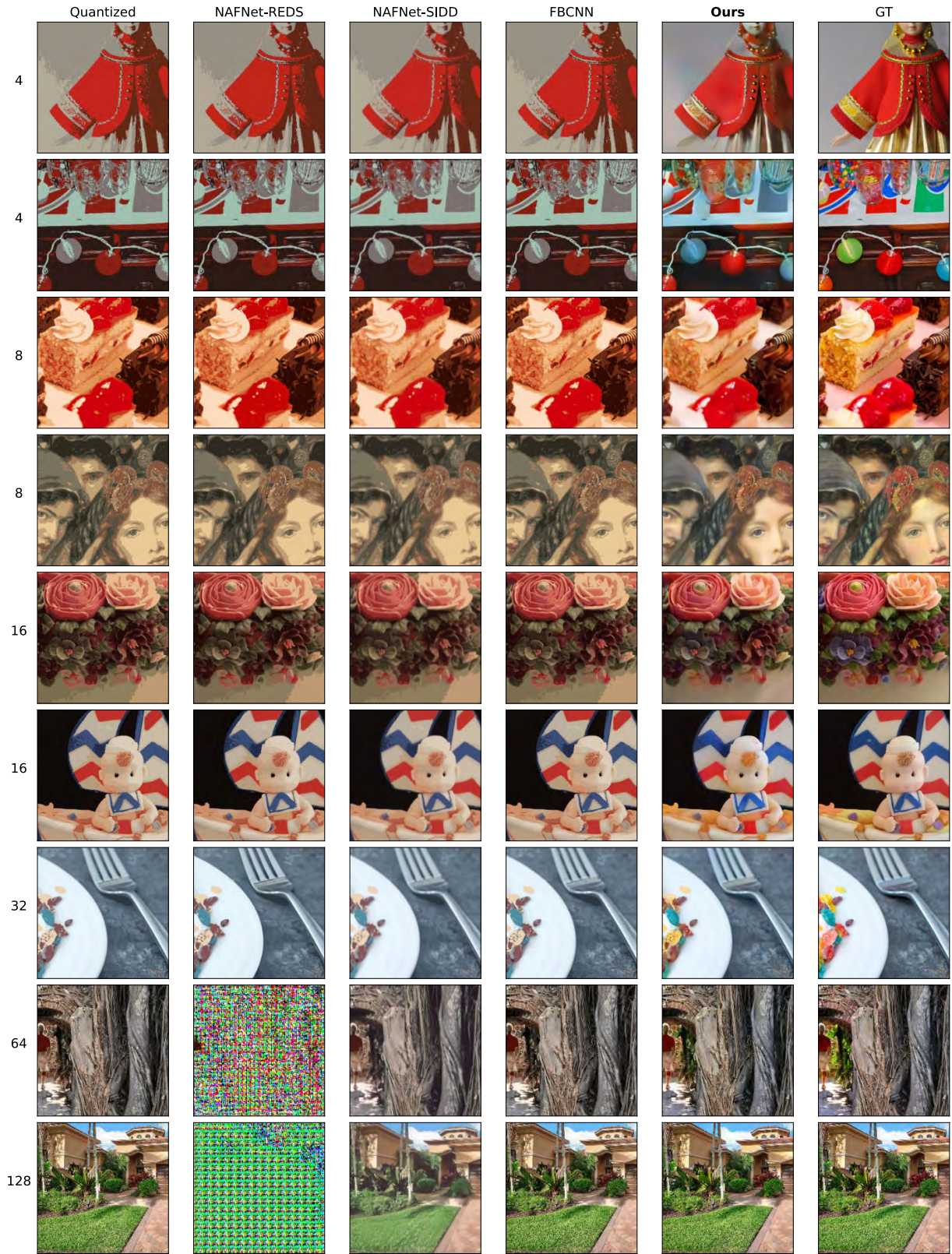
Figure 5. Dequantization without texture conditioning - only the quantized image is the input (with number of colors specified in the row). Across the board, our method (**fifth column**) produces more natural and aesthetic results as compared with baseline denoisers [2, 15].

4. Results & Discussion

Dequantization Quality: Core to our method is measuring how well our diffusion models can dequantize natural images. To evaluate, we quantize 1000 test images at several palette sizes, and measure how well our model can predict the source GT image. As shown in Fig. 3, our model outperforms all baselines, which suggests a specialized dequantization model is warranted. Shading indicates standard error bars. Fig. 5 shows qualitative evaluation. Our method is able to smooth harsh quantization boundaries and restore desaturated colors to a natural level. Our method succeeds because it was specifically trained to restore quantized images; existing denoisers are able to restore JPEG noise but not the aggressive decimation from quantization.

Dequantization with Texture: In Fig. 4, we evaluate our models when texture conditioning is available. **(top)** We are quantizing the GT image and attempting to reconstruct it from its palette, along with additional texture conditioning (i.e. thresholded gradient, gradient, luminance). **(bottom)** In the case of palette transfers, we do not know the GT image; thus to evaluate, we can quantize the generated image and compare it (i.e. calculate the SSIM/PSNR) with the input (which is a quantized image). Observe how the quality improves with more colors in the palette. Thus, the Palette-SSIM metric is a reasonable proxy when GT is not available, but it's not perfect. Notice that ours − T (thresholded gradient) performs better than ours − L − post (which conditions on GT luminance and enforces it as post-process). Palette error has curious difficulties that follow from the behavior of the quantization algorithm. If we impose the original image's luminance, we might change the quantization substantially. In Fig. 6 we evaluate our four methods when texture accompanies the quantized image as conditioning. The row indicates the number of colors in the palette; the first column is the quantized image. The subsequent 3 columns show conditioning texture for the 3 models. ours − L − post and ours − L condition on luminance, with the former enforcing luminance as a post-process. ours − G conditions on image gradients and ours − T conditions on thresholded gradients. All models - even those with less information available - are capable of dequantizing the input quite well, though the luminance-based models tend to achieve better color reproduction.

Palette Transfer: A key motivation for good image dequantization is palette based image editing. In Table 1, we evaluate using our palette error metric (because GT is not available). The palette is transferred from a random test image via minimum bipartite graph matching for selecting colors, where most similar color is the criteria (`color` method). Note that error metrics are generally worse with texture conditioning - this is because with texture, the network must find solutions that respect both the palette and texture that was asked for, which results in deviations from

Method	Texture On				Texture Off			
	8 colors		32 colors		8 colors		32 colors	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
ours − T	**23.0**	**0.774**	**27.4**	**0.874**	25.2	**0.842**	28.8	0.901
ours − G	17.8	0.516	19.1	0.538	25.6	0.834	28.5	0.893
ours − L	16.3	0.526	15.8	0.490	**25.7**	0.839	**29.1**	**0.905**
ours − L − post	13.2	0.481	12.8	0.444	12.8	0.468	12.6	0.457

Table 1. Palette transfer metrics for `color` method. Even though ours − T performed slightly worse in texture-conditioned dequantization (Fig. 4), it excels in palette transfers because thresholded gradients decorrelate local texture from local luminance.

the palette. When the palette is the only conditioning, it's easier to respect it. Thresholded gradient (ours − T) generalizes the best when texture is supplied. All methods succeed when texture is not supplied, because the conditioning is identical amongst the models in those settings (except ours − L − post, where luminance is not an input but is still enforced as a post-process). We present qualitative palette transfer results in Fig. 8. Five rows of `color` method and five rows of `negative − color` are shown. ours − T excels in generating natural results that respect both the texture and requested palette, because the texture conditioning is better decorrelated with the palette conditioning.

Inpainting: Because our method relies on a pretrained diffusion model, and it can map quantized image patches to natural images, we can present another application: color-conditioned image inpainting. We show qualitative results in Fig 7. The first column shows the input with a patch recolored with: (first 3 rows) the mean color of its support or (bottom 3 rows) a random color. The Mask column shows the affected region in black. The next three columns show the result of quantizing: the input; our luminance-conditioned result; and our no-texture-conditioned result. ours − L and ours − noTex are generated results. The final column is the ground truth image. In the first three rows, our method respects the requested color and texture fairly well, though the absence of texture (col. 7) causes some smoothing. Row four shows the problem with luminance conditioning: ours − L is incapable of generating a bright yellow color since that is different from the dark luminance in that patch. The texture and color conditioning conflict with each other. Rows 5 and 6 show similar outcomes. Generally, luminance conditioning results in pleasing and well-harmonized images that do not always respect the exact color that was asked for. Removing texture conditioning trades-off local shading reproduction and harmonization in exchange for color accuracy. In the absence of ground truth recolored images (final 3 rows), measuring palette error between **Q(Ours)** and **Q(Input)** provides an indication of whether the user got what they asked for.

Conclusion: We have presented an image dequantization procedure and demonstrated it can perform extreme palette transfers. We proposed several foundation models with slightly different conditioning and trade-offs. In practice, all methods should be available in an artist's toolbox.
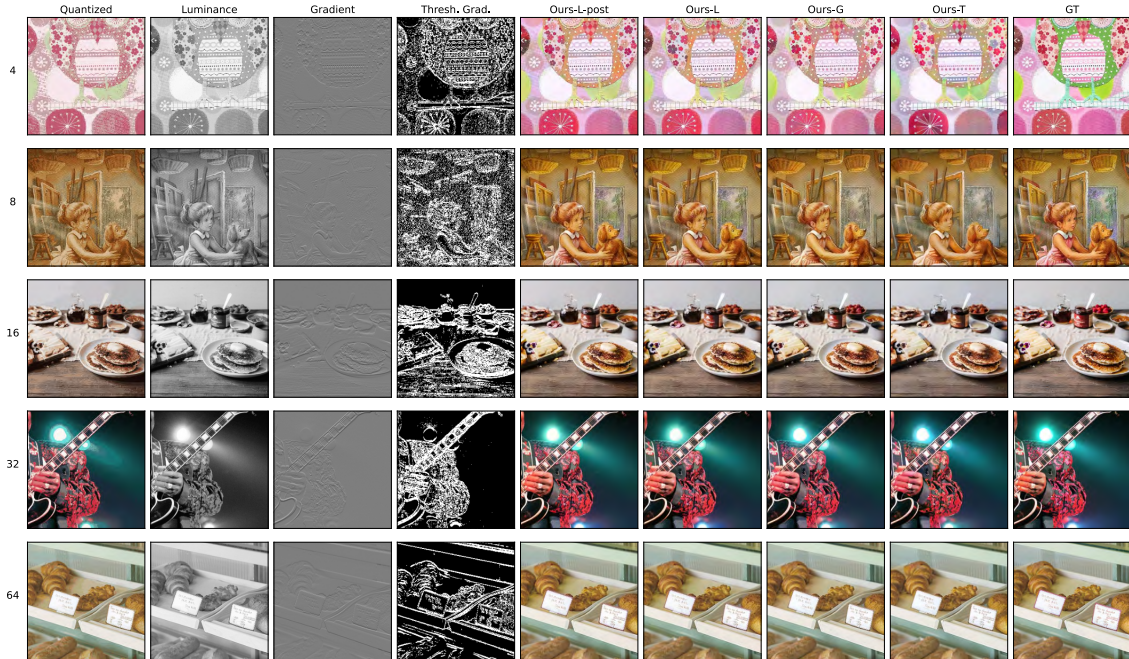
Figure 6. Qualitative evaluation of dequantization when texture is available. All methods are generally successful, though methods with luminance available (ours − L − post, ours − L) naturally perform better.
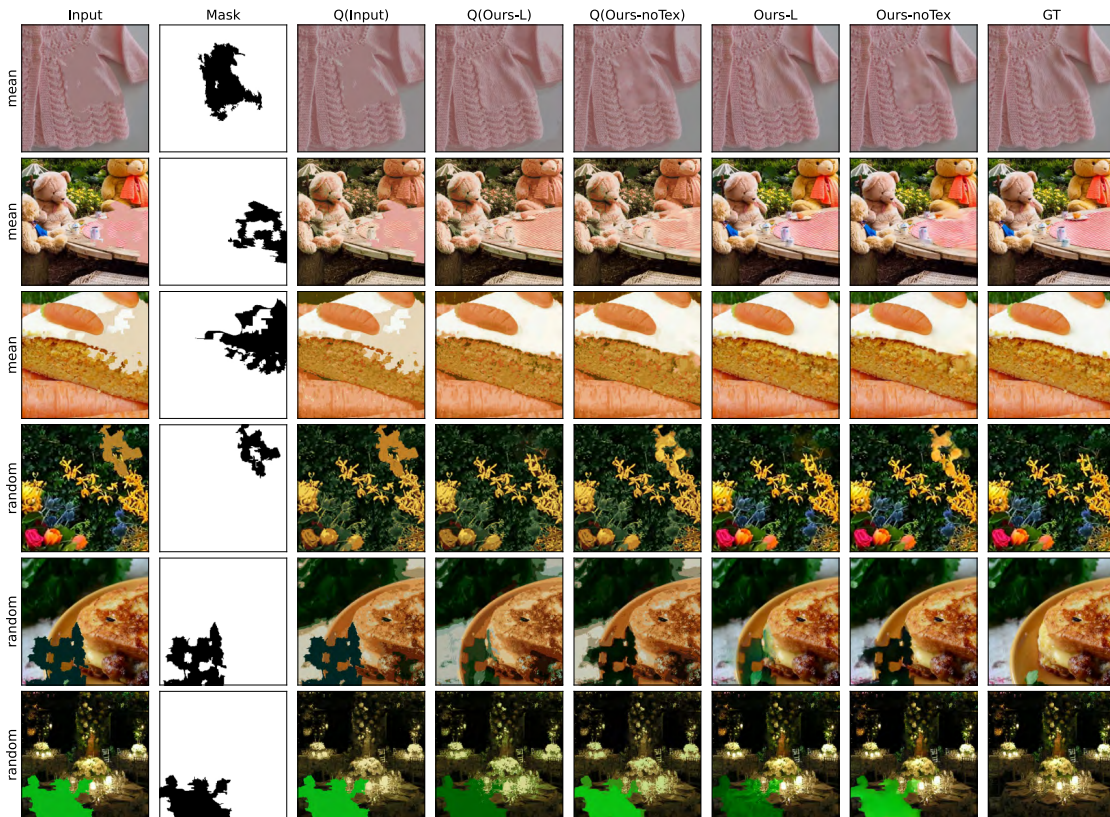

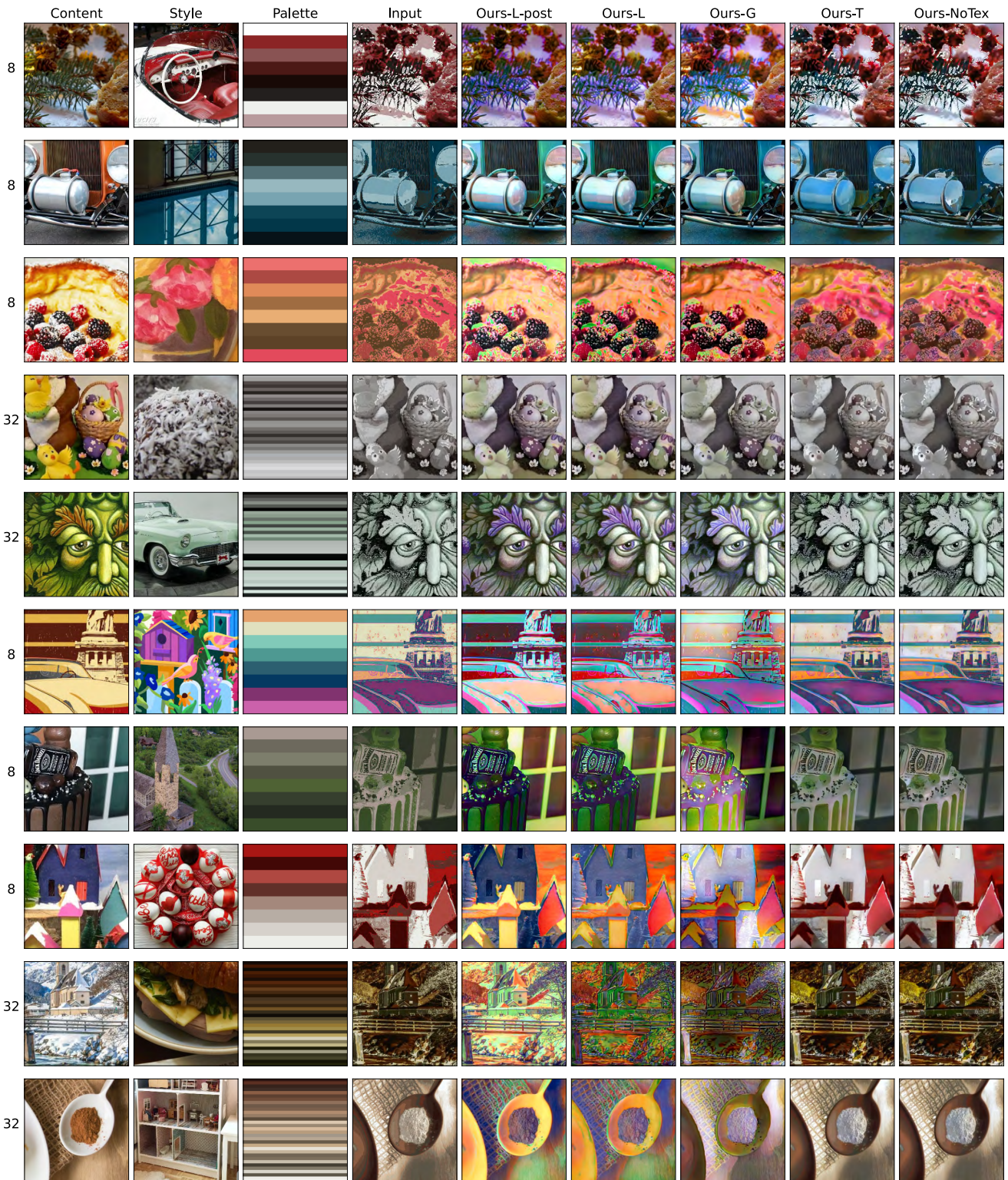Figure 7. Our image dequantizer can inpaint with color and/or texture conditioning, a useful artist edit.

Figure 8. Using color palettes as a form of abstraction, we can transfer color from one image ($\mathbf{2^{nd}}$ **column**) to another ($\mathbf{1^{st}}$ **column**). The extracted palette is shown in the $\mathbf{3^{rd}}$ **column**, and the quantized input to our method is shown in the $\mathbf{4^{th}}$ **column**. The remaining five columns show the synthesized results. Models with a weaker notion of texture (last two columns) can better respect the requested palette.

References

[1] Kalliopi Basioti and George V. Moustakides. Image dequantization using generative models as priors, 2020. 2

[2] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022. 3, 5

[3] Junho Cho, Sangdoo Yun, Kyoung Mu Lee, and Jin Young Choi. Palettenet: Image recolorization with given color palette. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 2

[4] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr^2: Image style transfer with transformers, 2022. 2

[5] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021. 1, 2

[6] Omar Elharrouss, Noor Almaadeed, Somaya Al-Maadeed, and Younes Akbari. Image inpainting: A review. *Neural Processing Letters*, 51:2007–2028, 2020. 2

[7] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *arXiv preprint arXiv:2012.09841*, 2021. 2, 4

[8] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. 2

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 2

[10] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network, 2017. 2

[11] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 2

[12] Youngbae Hwang, Joon-Young Lee, In So Kweon, and Seon Joo Kim. Color transfer using probabilistic moving least squares. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3342–3349, 2014. 2

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 2

[14] Jaeseok Jeong, Mingi Kwon, and Youngjung Uh. Training-free style transfer emerges from h-space in diffusion models, 2023. 2

[15] Jiaxi Jiang, Kai Zhang, and Radu Timofte. Towards flexible blind jpeg artifacts removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4997–5006, 2021. 1, 3, 5

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2018. 2

[17] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020. 2

[18] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 2

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*, 2020. 2

[20] Siavash Khodadadeh, Saeid Motiian, Zhe Lin, Ladislau Boloni, and Shabnam Ghadar. Automatic object recoloring using adversarial learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1488–1496, 2021. 2

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1, 2, 21

[22] Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Bjorn Ommer. Content and style disentanglement for artistic style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4422–4431, 2019. 2

[23] Zhexin Liang, Zhaochen Li, Shangchen Zhou, Chongyi Li, and Chen Change Loy. Control color: Multi-modal diffusion-based interactive image colorization. *arXiv preprint arXiv:2402.10855*, 2024. 2

[24] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *The European Conference on Computer Vision (ECCV)*, 2018. 2

[25] Shiguang Liu. An overview of color transfer and style transfer for images and videos. *arXiv preprint arXiv:2204.13339*, 2022. 2

[26] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4990–4998, 2017. 2

[27] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. *arXiv preprint arXiv:2201.09865*, 2022. 1, 2

[28] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2022. 1

[29] F. Pitie, A.C. Kokaram, and R. Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1434–1439 Vol. 2, 2005. 2

[30] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001. 2

[31] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017. 2

[32] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 4

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2, 3, 4

[34] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. StyleGAN-T: Unlocking the power of GANs for fast large-scale text-to-image synthesis. In *International Conference on Machine Learning*, volume abs/2301.09515, 2023. 2

[35] Mengjun Tao and Richard Jiang. Dual-Domain Diffusion based Progressive Style Rendering Towards Semantic Structure Preservation. 1 2023. 2

[36] Vaibhav Vavilala and David Forsyth. Controlled gan-based creature synthesis via a challenging game art dataset - addressing the noise-latent trade-off. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3892–3901, January 2022. 2

[37] Vaibhav Vavilala and David Forsyth. Convex decomposition of indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9176–9186, 2023. 2

[38] Vaibhav Vavilala, Seemandhar Jain, Rahul Vasanth, Anand Bhattad, and David Forsyth. Blocks2world: Controlling realistic scenes with editable primitives, 2023. 2

[39] Vaibhav Vavilala, Florian Kluger, Seemandhar Jain, Bodo Rosenhahn, and David Forsyth. Improved convex decomposition with ensembling and boolean primitives, 2024. 2

[40] Vaibhav Vavilala and Mark Meyer. Deep learned super resolution for feature film production. In *ACM SIGGRAPH 2020 Talks*, SIGGRAPH '20, New York, NY, USA, 2020. Association for Computing Machinery. 2

[41] Vaibhav Vavilala, Rahul Vasanth, and David Forsyth. Denoising monte carlo renders with diffusion models, 2024. 3

[42] F. Wu, W. Dong, Y. Kong, X. Mei, J.-C. Paul, and X. Zhang. Content-based colour transfer. *Computer Graphics Forum*, 32:190–203, 2013. 2

[43] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22428–22437, 2023. 2

[44] Serin Yang, Hyunmin Hwang, and Jong Chul Ye. Zero-shot contrastive loss for text-guided diffusion image style transfer, 2023. 2

[45] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 1, 2, 3

[46] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017. 2

[47] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models, 2023. 2