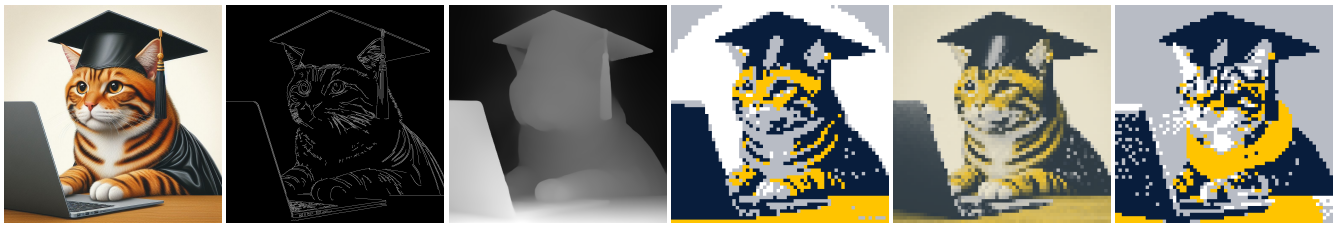
Supplementary Material: SD-πXL: Generating Low-Resolution Quantized Imagery via Score Distillation

ALEXANDRE BINNINGER, ETH Zurich, Switzerland
OLGA SORKINE-HORNUNG, ETH Zurich, Switzerland



input | Canny edge | depth map prediction | initialization (argmax) | initialization (softmax) | result

Fig. 1. Initial setup for our parameter comparisons. The sequence includes the original input image, its corresponding Canny edge detection [Canny 1986], and depth map prediction via DPT [Ranftl et al. 2021]. We also show the initial state of the generator, using the softmax- and argmax-generation modes, and the result using our default parameters. The used prompt is "A cat wearing a graduation hat using a computer."

Our supplementary material offers further insights into the implementation of our method, illustrates the impact of various parameters, and includes details about the evaluation of our approach.

CCS Concepts: • **Computing methodologies** → **Image processing**; *Image representations*; • **Applied computing** → Fine arts.

Additional Key Words and Phrases: pixel art, image processing

ACM Reference Format:
Alexandre Binninger and Olga Sorkine-Hornung. 2024. Supplementary Material: SD-πXL: Generating Low-Resolution Quantized Imagery via Score Distillation. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3680528.3687570

1 IMPLEMENTATION DETAILS

This section outlines the specifics of our method's implementation, including the default parameters and technical configurations utilized. The source code is made available at https://github.com/AlexandreBinninger/SD-piXL.

1.1 Technical details

Our method employs backpropagation through the encoder of the latent diffusion model. To enhance efficiency and minimize memory usage, we utilize a distilled version of the stable diffusion VAE, namely taesdxl [Madebyollin 2023]. We adopt mid versions of the Canny edge and depth ControlNets [Zhang et al. 2023], specifically "controlnet-canny-sdxl-1.0-mid" and "controlnet-depth-sdxl-1.0-mid" [von Platen et al. 2022]. This choice strikes a balance between computational resource demands and the effectiveness of spatial conditioning. An aspect of our generator g is its invariance to translation by a constant across the last channel due to the softmax operation. Indeed, given a constant a, the relation $g(\theta) = g(\theta + a)$ holds because $\frac{e^{\lambda_{i,j,k}+a}}{\sum_{l=0}^{n-1} e^{\lambda_{i,j,l}+a}} = \frac{e^{\lambda_{i,j,k}}}{\sum_{l=0}^{n-1} e^{\lambda_{i,j,l}}} = \pi_{i,j,k}$. Therefore, we center the weights around zero in every iteration, ensuring they remain within a reasonable range to avoid floating point precision issues.

1.2 Parameters

The execution of our method typically involves 6000 to 10000 epochs, translating to a runtime of approximately 1.5 to 2.5 hours on an Nvidia RTX 4090 GPU. The standard parameters are set as follows: the temperature parameter of the Gumbel-Softmax reparameterization τ is fixed at 1, the guidance scale s at 40, and the FFT loss weight w_{FFT} at 20. Unless specified otherwise, the default image size for the presented results is 64×64, and the norm used for initializing the parameter of the generator is the L_1 norm (see main paper, Sec. 4.2). We use PyTorch [Paszke et al. 2019], with the generator weights optimized using the AdamW optimizer [Loshchilov and Hutter 2019]. The optimization process follows a constant learning rate of 0.25, starting with a warm-up phase of 250 steps. Although backpropagation through an argmax function can be realized by duplicating the gradient from the softmax operation [Esser et al. 2021], our method primarily employs the softmax function. To approximate an argmax-like behavior, we can reduce the value of τ, see details in a comparative analysis in Sec. 2.4.

Image augmentations are applied randomly with the following probabilities: grayscale (0.2), horizontal flip (0.5), perspective distortion (0.5) with a distortion scale of 0.3. The scales for both the Canny edge and depth map ControlNet conditioning are set uniformly at 0.35. We apply Gaussian blur with a radius of 1 pixel to the Canny edge detection for smoothing effects. Finally, the uniform sampling of the parameter $t \sim \mathcal{U}(a, b)$ starts with $a = 20$ and $b = 980$, with b linearly decreasing to 800 at the midpoint of our method's execution, and staying constant afterwards. This time step annealing strategy is inspired from [Yu et al. 2023].

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687570

2 PARAMETER COMPARISONS

In this section, we conduct an indicative comparison to justify the choice of our default parameters. We also propose to analyze how each parameter can be combined to achieve different effects.

2.1 Input

For our experiments, we decided to use an image of a cat generated via a state-of-the-art diffusion model [Betker et al. 2023] as input. Fig. 1 shows the input image, its Canny edge and depth map prediction. We also show the initialized state of the generator, both with the argmax- and softmax-generation, as explained in Sec. 5.1 in the main paper. We also showcase the result obtained with our default parameters.

2.2 Loss functions

We recall that our loss function can be written as a sum of three terms:

$$\nabla_\theta \mathcal{L} = \nabla_\theta \mathcal{L}_{Noise} + s\nabla_\theta \mathcal{L}_{Sem} + w_{FFT}\nabla_\theta \mathcal{L}_{FFT}. \tag{1}$$

We analyze the noise and semantic loss together, as they come from the latent score distillation sampling term, and the smoothness loss separately.

2.2.1 Noise loss and semantic loss. To show the respective role of the noise loss $\nabla_\theta \mathcal{L}_{Noise}$ and the semantic loss $\nabla_\theta \mathcal{L}_{Sem}$, we perform an ablation by setting $w_{FFT} = 0$ and varying the guidance scale s in Fig. 3. We notice that when the semantic loss s is too small, the resulting final image is smoothed out. In contrast, too high values of the guidance scale leads to noisier and saturated results. While the semantic loss is responsible for the semantics-awareness of our optimization, the noise loss acts as a variance reduction term [Poole et al. 2022]. The visualization in Fig. 2 corroborates this argument, showing that the semantic loss is responsible for the semantic details, while the noise loss prevents noisy sampling, especially for the lower time steps. The visualization is performed by simply using the latent decoder of the latent diffusion model [Madebyollin 2023] on the gradient of the loss functions.

2.2.2 Smoothness loss. The fast Fourier transforms serves the purpose of smoothing out the resulting image during the optimization process. While not central to our technique, it tends to improve the result and show that any classic loss can be used to redirect the generation of SD-πXL according to any objective. We demonstrate its influence in Fig. 4 by varying the weight w_{FFT}. We can see that the higher the weight, the fewer details appear in the generated image. While oversmoothed results are generally not desirable, the smoothness loss can act as a stylistic parameter for image abstraction purposes.

2.3 Augmentation

In our optimization, we randomly apply a grayscale filter and a perspective transformation. We show the results of applying these filters at different frequencies.

2.3.1 Grayscale. In Fig. 5 we show a comparative demonstration of varying the probability (or the frequency) that the augmentation performs a grayscale filter before being fed to the denoiser. Interestingly, we can notice that applying a grayscale filter tends to tone down the color distribution of the output. When no grayscale filter is applied ($p_{grayscale} \approx 0$), the color tends to be saturated. Conversely, frequent use of the grayscale filter ($p_{grayscale} \approx 1$) restricts the optimization process to a monochromatic view of the input palette. In this scenario, the diffusion model only perceives color based on their contribution to the grayscale tones, leading to the prominence of yellow blotches in our example.

2.3.2 Random perspective. The perspective transform simulates the effect of viewing the image from different angles. This transform is useful, as it introduces variation in the generated images, forcing the optimization to generalize its semantic update of features in images irrespective of their orientation or angle. The impact of this transformation is illustrated in Fig. 6, where we observe that increased distortion scales lead to outputs more aligned with spatial conditioning, as the optimization is required to adjust its updates more broadly. However, excessively high distortion scales introduce undesired noise into the final image. In our experiments, we demonstrate the effects using a 1.0 probability for applying the perspective transformation during optimization. However, in practical applications, we balance its impact by not applying random perspective transformations in every instance.

2.4 Temperature parameter τ

The Gumbel-softmax is parameterized by a scalar τ that modulates its proximity to a categorical distribution [Jang et al. 2017; Maddison et al. 2017]. In Fig. 7, we investigate the assertion that τ should remain within certain limits. As discussed in Sec. 4.1 of the main paper, lower values of τ enhance the resemblance of Gumbel-softmax to categorical sampling, but this leads to an increase in noise in the generated images. Even though our loss integrates a noise-preservation component, too noisy generated images result in imprecise parameter updates. Conversely, higher values of τ cause the Gumbel-softmax to approximate a uniform distribution, resulting in images with colors that are more uniform and smooth, with diminished semantic clarity. Although the optimization process is semantically driven and continues to adjust the generator parameters to align the image with the input prompt, excessively high values of τ can compromise the optimization's ability to capture color nuances. It is noteworthy that τ values in the range of 0.25 to 2 yield aesthetically pleasing results, although with distinct stylistic differences.

2.5 Initialization

Initialization can be important for generative techniques based on score distillation [Jain et al. 2023]. We show its effect in Fig. 8. Without the use of ControlNet [Zhang et al. 2023], the optimization process lacks access to spatial information from the input image, resulting in outcomes that can deviate from the original image.

2.6 ControlNet weights

In Fig. 13, we explore the impact of varying weight parameters within the ControlNet framework [Zhang et al. 2023]. This analysis reveals a natural progression: the absence of ControlNet results in

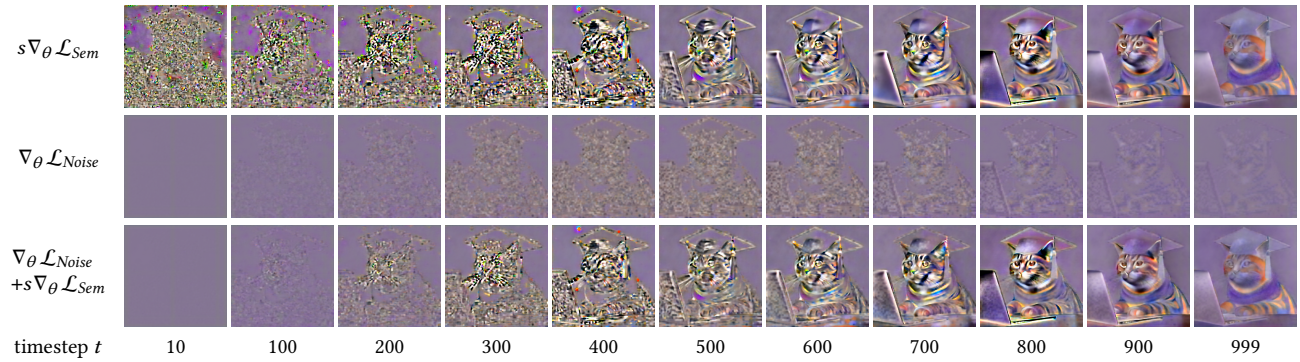


Fig. 2. Illustration of the latent score distillation sampling loss with fixed guidance scale $s = 40$. The first row represents the semantic loss, while the second row is the noise loss. The full latent score distillation sampling loss $\mathcal{L}_{LSDS}$ is shown on the third row. Note that this illustration is computed by using the latent decoder of the latent diffusion model on the loss gradient, which is not a linear operation.

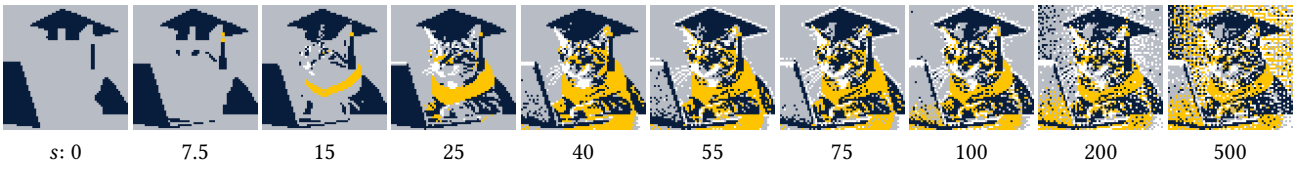


Fig. 3. Illustration of the influence of the guidance scale s. For this experiment, the FFT loss was not used, i.e. $w_{FFT} = 0$.

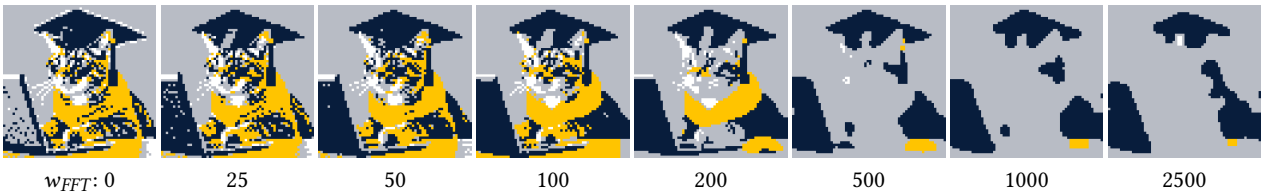


Fig. 4. Illustration of the influence of the Smoothness Loss $\mathcal{L}_{FFT}$ with increasing smoothness loss weight w_{FFT} on the final image.

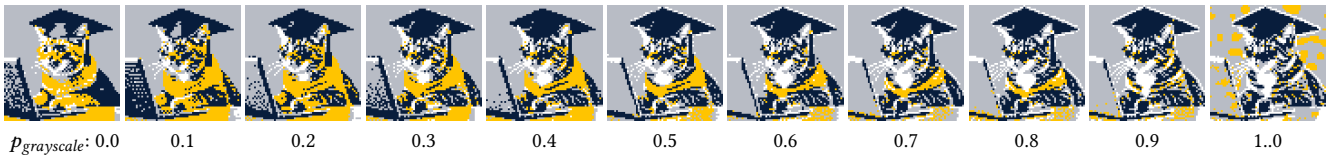


Fig. 5. Depiction of how varying the probability of applying grayscale augmentation during the augmentation phase impacts our optimization process.

outputs that are less faithful to the original input, which demonstrates the role of ControlNet in balancing strict adherence to the initial input spatial conditioning. Interestingly, even without explicit ControlNet guidance, the initialization phase of our method inherently allows for a degree of control that influences the final outcome, as shown in Fig. 8.

3 EVALUATION

In this section, we show several comparisons and evaluations of SD-πXL with different pixelization methods. First, we propose a visual comparison over several images with diverse styles. Second, we present a quantitative evaluation to assess the pixelization quality through three indirect measures, namely how the result image aligns with the input prompt (semantic), how the result image is alike the input image (fidelity), how aesthetically pleasant the result image looks (aesthetics). Finally, we present the results of a perceptual study over 56 participants that further corroborates the results of the quantitative evaluation.

3.1 Visual comparison

Our visual comparison includes pictures of common food and of a person, as well as a video game sprite and a painting. The resolution of each image is reduced by a factor of eight during the pixelization process. Results are shown in Fig. 9. We compare with various

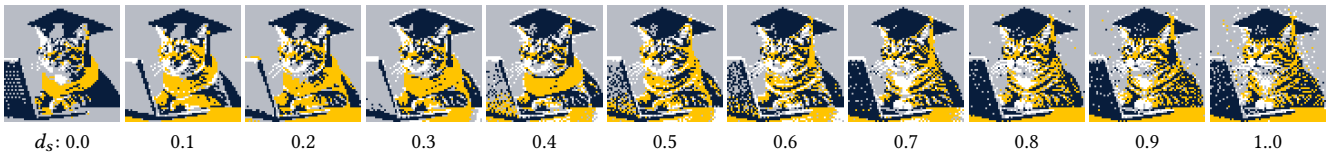


Fig. 6. Illustration of the influence of the perspective transformation distortion scale. To better show the influence of the random perspective, the probability of perspective transformation during the augmentation is set to 1.0 for this experiment.

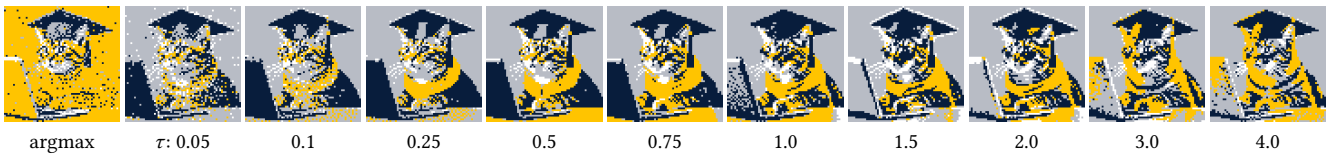


Fig. 7. We compare the effect of varying the magnitude of the temperature τ parameterizing the Gumbel-softmax operation. For the first column, we use an argmax operation during the optimization and copy the gradient from the softmax operation for backpropagation.

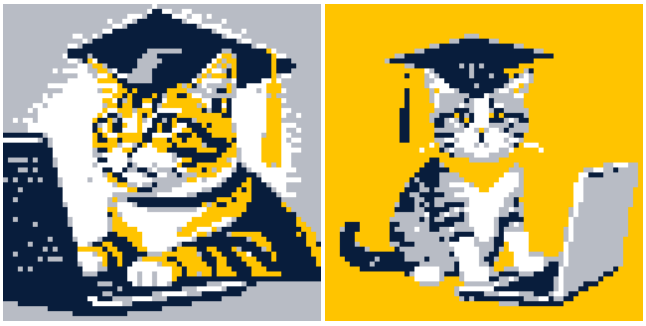


Fig. 8. We compare the effect of initializing the network based on the input image with random initialization of the generator parameters. The results are computed without ControlNet conditioning, to better showcase the role of initialization in our method.

pixelization techniques, namely Pixelated Image Abstraction (PIA) [Gerstner et al. 2012], Deep unsupervised pixelization (DUP) [Han et al. 2018], Make Your Own Sprites (MYOS) [Wu et al. 2022], and VectorFusion [Jain et al. 2023].

Table 1. Evaluation through a perceptual study, highlighting the performance of our method (Ours) in comparison to PIA and VectorFusion through semantic, fidelity and aesthetics questions. Each column aggregates the rankings across all questions in a specific category, representing the percentage of participants who placed each method at the respective rank (1, 2, 3, or 4) for that category.

Method	Semantic				Fidelity				Aesthetics			
Rank	1	2	3	4	1	2	3	4	1	2	3	4
PIA	24.6	23.0	31.2	21.2	**49.3**	37.9	11.1	1.8	25.5	26.0	33.1	15.5
VectorFusion	22.0	15.0	16.1	46.9	0.8	0.8	8.0	90.4	17.5	12.1	19.5	50.8
Ours-*K-means*	**36.2**	37.0	22.6	4.2	47.1	48.0	4.6	0.2	26.5	38.8	22.3	12.4
Ours-*palette*	17.1	25.0	30.1	27.7	2.7	13.3	76.3	7.6	**30.5**	23.1	25.1	21.3

Nearest-neighbor interpolation assigns the color value of the nearest original pixel to each pixel in the downscaled image, often resulting in a blocky or pixelated appearance. This method does not produce a color-quantized image, and its lack of semantic and geometric awareness can impair readability. In contrast, PIA [Gerstner et al. 2012] improves results through spatial and color space clustering, but its lack of semantic understanding can sometimes result in images that are difficult to interpret, such as the character's face and tie.

We also explore state-of-the-art neural techniques. DUP [Han et al. 2018] is retrained on its own dataset, and we use pretrained weights for MYOS [Wu et al. 2022]. For MYOS, we noticed that downscaling the input to divide the size of the image by two prior to using the neural network with cell size of 4 yields better results than directly using a cell size of 8. We therefore decided to present this output in our visual comparison. The primary content of the dataset used for training these neural methods is clip art images, which introduces a domain generalization issue. For example, DUP's outputs on photographic images tend to be overly saturated. MYOS, on the other hand, does not precisely downscale but rather employs "cell-aware" techniques based on a cell size of 4, as described in [Wu et al. 2022], which can lead to uneven pixelization in practice. This also constrains the pixelization to a finite range (2 to 8 in the case of MYOS). As these methods do not enforce a uniform color within

Table 2. First quartile (Q1), median (Med.) and interquartile range (IQR) of the results of our perceptual study, according to semantic similarity, fidelity to input image and aesthetic appeal.

Method	Semantics			Fidelity			Aesthetics		
	Q1	Med.	IQR	Q1	Med.	IQR	Q1	Med.	IQR
PIA	2.0	3.0	1.0	1.0	2.0	1.0	1.0	2.0	2.0
VF	2.0	3.0	2.0	4.0	4.0	0.0	2.0	4.0	2.0
Ours-*K-means*	1.0	2.0	2.0	1.0	2.0	1.0	1.0	2.0	2.0
Ours-*palette*	2.0	3.0	2.0	3.0	3.0	0.0	1.0	2.0	2.0

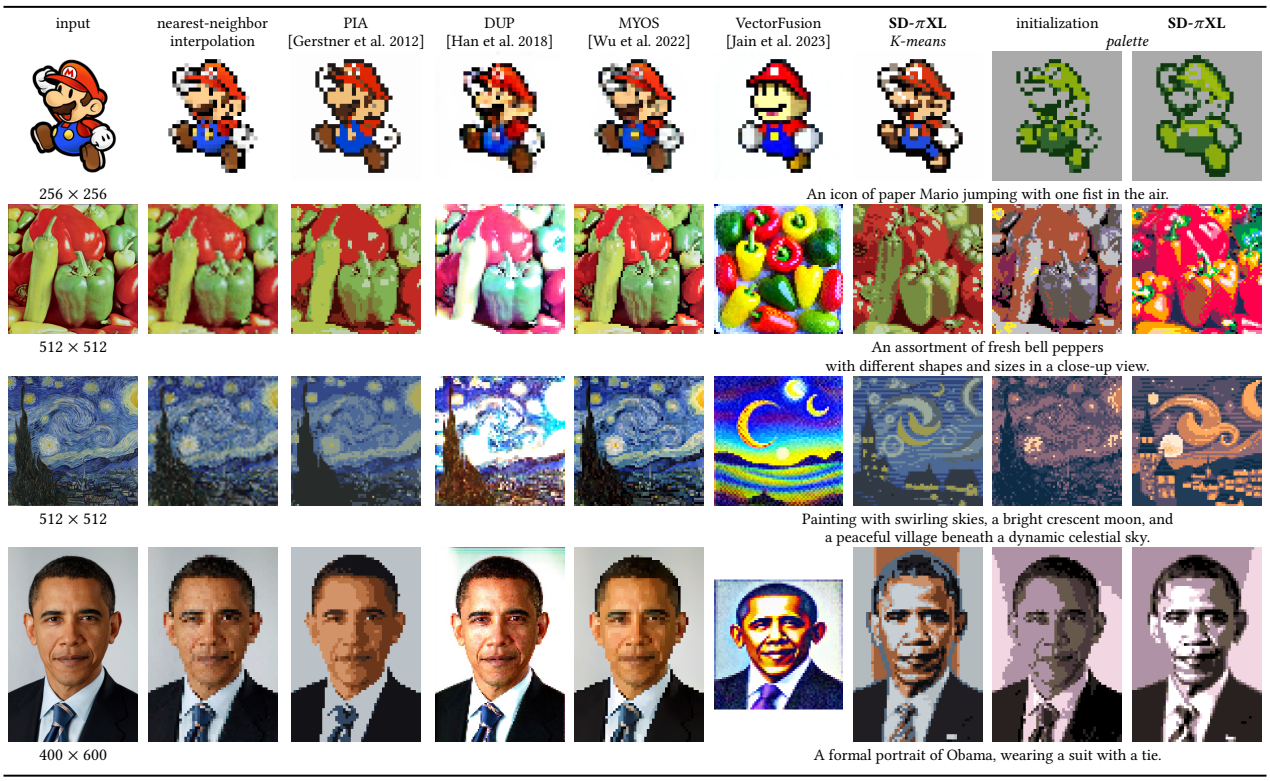


Fig. 9. Visual comparison of various pixelization methods applied to images downscaled by a factor of 8. The input images are displayed on the left, with their sizes indicated below. Both VectorFusion and SD-πXL are initialized with the input image as their initial state, and conditioned on the prompt indicated below their respective results. Due to VectorFusion's limitation with non-square images, a 64×64 grid is employed for the examples in the last row. SD-πXL is demonstrated in two variations: one utilizing *K-means* for color palette determination, and the other employing a challenging 5-color *palette*. We also show the initialization with the palette, to demonstrate how our method differs from classic palette matching. While PIA and the K-means variant of SD-πXL operate within a 8-color limit, nearest-neighbor interpolation, DUP, MYOS, and VectorFusion have no such constraints on their color palettes and are not quantized. (The paper Mario character is ©Nintendo Co., Ltd.)

each cell, they ultimately rely on nearest-neighbor downsampling to achieve a distinct pixelized effect.

VectorFusion [Jain et al. 2023] segments the image into an even square grid and employs a differentiable vector graphics renderer [Li et al. 2020]. The technique uses score distillation to optimize the renderer parameters, specifically the colors of each cell. Contrary to our method, the color space of VectorFusion output is never constrained. We observe that the output tends to be saturated, suggesting that employing a color palette mitigates the saturation effect typically associated with score distillation methods. Starting with the input image as a base, results bear a general resemblance to the original, yet exhibit notable divergence in certain areas. This is visible in all examples, particularly in the positioning of the bell peppers, the global composition of the painting, and the game sprite's positioning (Fig. 9). This legitimates the use of ControlNet [Zhang et al. 2023] to encourage more similarity with the input image, which is further validated by the outcomes achieved with SD-πXL.

We introduce two variations of our method: the first employs K-means to create a 8-color palette through color space clustering. The second utilizes challenging color palettes, and to demonstrate the distinctiveness of our approach, we also present its initialization. This illustrates the progression from a basic palette association to the final output. Moreover, it shows that a simple downscaling followed by a palette transfer does not provide convincing results, justifying our approach based on an optimization process using an image generator constrained to the color palette. SD-πXL consistently delivers outputs that more closely mirror the input image than VectorFusion, while effectively capturing the semantic core of the prompt. This is particularly visible in the case of the painting (third row of Fig. 9), where the overall structure of the original image is kept in its pixelized version, but with some divergence in details that conveys meaning more aligned with the input prompt. On this specific example, MYOS achieves a result that is closer to the input image, and VF a result that aligns more to the input prompt, while our method reaches a middle ground between both objectives. Remarkably, this equilibrium is maintained despite the constraints of a limited color palette and substantial color variations in the input image.

3.2 Quantitative evaluation

We present a quantitative evaluation of pixelization methods, which assesses fidelity to the input image, semantic similarity and aesthetics qualities of different pixelizations methods. To that effect, we first sample 150 prompts from PartiPrompts [Yu et al. 2022]. We divide them equally into 3 categories for different target sizes: 32×32, 48×48, and 64×64. Each set of 50 prompts is sampled according to this allotment: 10 from the 'Animal' category, 10 from the 'People' category, 10 from the 'Outdoor Scenes' category, 5 from the 'Indoor Scenes' category, 5 from the 'Food & Beverages' category, 5 from the 'Vehicles' category, and 5 from the 'Produce & Plants' category. We order the prompts by length, associating the shorter prompts to the smallest size, and the longer, more detailed ones to the 64×64 category. To avoid recurrence of the same concept (e.g., the same animal), we resort to a large language model (LLM) [Brown et al. 2020] to further filter out repetitive prompts. In some instances, we remove color mentions, to avoid further bias of the results, favoring methods that do not offer a quantization of the color space. Subsequently, we produce input images corresponding to these prompts using SDXL [Podell et al. 2023], employing a CLIPScore-based [Radford et al. 2021] rejection sampling technique. This involves generating six images and selecting the one with the highest CLIPScore. In instances where SDXL fails to produce an image of satisfactory quality, we turn to DALL·E 3 [Betker et al. 2023] for additional image generation. The prompts and generated images can be found in the supplementary material.

We quantitatively compare PIA [Gerstner et al. 2012], quantized MYOS [Wu et al. 2022], and VectorFusion [Jain et al. 2023] with three variants of our method. The *palettes* variant utilizes a palette randomly sampled from 34 options available on the Lospec website [los 2024]. The *K-means* variant employs a color palette determined through K-means clustering of the input color space, with both K-means and PIA using 8 colors for a fair comparison. The *adaptive* variant optimizes the color palette alongside the generator weights without Gumbel sampling, creating an output image that is not quantized and has an unconstrained color space, offering a fairer comparison with VectorFusion. This method also uses 8 colors to ensure the entire color space can be covered by a convex sum of input colors. MYOS only provides pixelization with integer factor. To compare with this method, we first downscale the input image with bicubic interpolation to four times the target output resolution, and then run MYOS with a cell size of 4, followed by nearest neighbour downscaling to the target output resolution. Because there is no integer pixelization factor for 48×48 images, we do not present results for this pixelization scale. We then apply two quantizations: a K-means color clustering with 8 colors, and a palette transfer with libimagequant [Lesiński 2024]. The palette used is the same as the palette of the corresponding image produced with SD-πXL-palette. Both VectorFusion and SD-πXL are run for 6000 steps and initialized with the input image. A selection of the results from SD-πXL with palettes and K-means on this dataset is shown in Fig. 14.

We use several metrics divided into three categories: semantic, fidelity and aesthetic. We show the results in Table 3 individually for each size, and the average over the 150 examples. The semantic evaluation includes CLIPScore L/14 similarity [Radford et al. 2021], determined by calculating the mean cosine similarity between the CLIP embeddings of the generated images and their respective text captions, and the human preference score [Wu et al. 2023], a model that can predict human preferences on prompt-images pairs. We can see that VectorFusion and SD-πXL-*adaptive* achieves the same score for semantic score, while SD-πXL-*K-means* and *palette* are slightly below, because of the constraints imposed by the color quantization. However, their CLIPScore tends to be higher than MYOS and PIA's, which are not driven semantically at all.

The fidelity to the input image is measured via the peak signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM) [Wang et al. 2004], two classic methods for image similarity measures [Horé and Ziou 2010]. On this metric, MYOS and PIA generally achieve the best score, as they are entirely driven by their similarity to the input images. Our SD-πXL-*K-means* closely follows. In contrast, the lack of spatial conditioning in VectorFusion can make the output greatly diverge from the input, lowering its fidelity score.

Although it is hard to quantitatively evaluate the aesthetics of an image, we use the LAION aesthetic classifier [Schuhmann 2023], a model specifically trained on human preference to assign aesthetic scores to images. We observe that the *palette* and *K-means* variants of SD-πXL outperform PIA, MYOS and VectorFusion on this aesthetics measure. Despite color quantization posing difficulties for semantic or fidelity analysis, the high aesthetic scores for SD-πXL-*palette* variant can be attributed to the harmonious color combinations, a feature that is lacking in VectorFusion, which performs poorly on this metric.

While all these metrics have a positive correlation with the quality of good pixelization, none of the metrics assesses directly the pixelization quality itself. To support this argument, we ran bilinear downscaling on our dataset and computed the metrics on the fidelity measure. While this technique is not well-adapted for pixelization purpose, the average PSNR is 16.8, and the average SSIM is 0.512, which are substantially higher than pixelization methods. This is expected, as bilinear interpolation is more faithful to the input image than pixelization method, but it shows that this measure alone is not an indicator of good pixelization. We interpret the results as such: CLIP-based methods like VectorFusion produce results that are more aligned semantically, MYOS and PIA produce the results that are the closest to the input, and SD-πXL achieves a satisfying compromise between the two different metrics, while being more aesthetically pleasing overall.

3.3 Perceptual study

To mitigate potential bias of the networks used to quantitatively evaluate our method, we conducted a comprehensive perceptual study to compare the effectiveness of PIA, VectorFusion, and SD-πXL with *K-means* and *palettes*. A total of 56 participants were recruited for the study. The survey presents participants with a series of images processed using the four different pixelization techniques.

3.3.1 Design. To design our perceptual study, we randomly sample 45 images from our quantitative evaluation dataset. Namely, we sample 15 images from each size category (32×32, 48×48, and 64×64). We further divide the perceptual study into three parts:

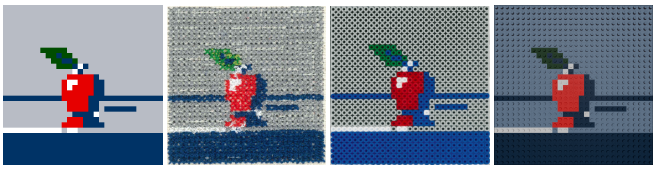


Fig. 10. SD-πXL generates low-resolution quantized images that are suitable for many fabrication applications, such as cross-stitch embroidery, fuse beads, or interlocking brick designs. The result image size is 30×30 pixels, generated without an initialization image, and only conditioned on the prompt "A red apple with a leaf on a blue table."

image fidelity, prompt similarity, and aesthetics appeal. Each part contains 5 images from each size category. The exact questions and layout are shown in Fig. 15, with the first question of each category. For each question, the order of the presented methods is randomized. We also show a visual representation of the answers in Fig. 15, essentially visualising as histograms the results reported in Table 1.

3.3.2 Results. Participants are asked to rank these 45 images based on specific criteria, such as semantic accuracy with respect to a prompt, fidelity to the input image, and aesthetic appeal. The perceptual study, as detailed in Table 1, offers a comparative evaluation of image pixelization methods across semantic, fidelity, and aesthetics categories. Notably, SD-πXL-*K-means* excels in semantic integrity, leading with 36.2% top-rank responses. PIA slightly dominates in fidelity, receiving a 49.3% first-place rating against 47.1% for SD-πXL-*K-means*. In aesthetics, SD-πXL-*palette* emerges as the preferred method with 30.5% top-rank responses, indicating its superior aesthetic appeal. In contrast, VectorFusion (VF) consistently ranks lower across all categories, suggesting limitations in semantic preservation, fidelity, and aesthetic appeal. The discrepancy between the perceptual study results and the quantitative evaluation for VectorFusion's semantic similarity performance is likely due to a bias of the scoring network towards non-quantized images.

To enhance the interpretability of our results, we showcase the first quartile, the median and the interquartile range in Table 2. The analysis reveals a varied landscape of participant responses. While PIA and SD-πXL-*K-means* demonstrate consistent performance across different categories, VF shows a higher variability in participant perceptions, particularly in the aesthetic domain: while it ranks last for more than half of the time, it also sporadically achieves good aesthetics appreciation. This variability suggests that user opinions on VF's performance are polarized. In contrast, SD-πXL-*palette* exhibits a balance between consistency and diversity in the feedback. Overall, the data indicates no single method excels in all aspects; each has unique strengths and weaknesses as perceived by the participants.

4 ADDITIONAL RESULTS

This section presents more crafted examples and experiments. We show how it is possible to use elements such as small images instead of pixel colors to produce mosaic-like effects.

Fig. 11. Additional examples of cross-stich embroideries showcased on . The sizes of the embroideries are as follow: first row is 32×32, second row is 48×48, and last row is 64×64.

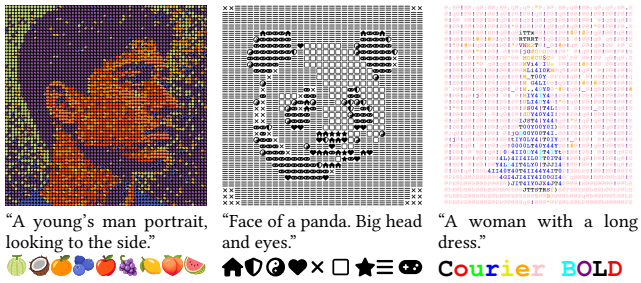


Fig. 12. SD-πXL demonstrates the ability to create mosaics using diverse elements, such as emojis or icons. Letters can be used to create compelling ASCII art. The results can be better seen on a screen with large zoom.

4.1 Additional fabrication

We present additional fabrication examples. Fig. 10 showcases an example of embroidery, fuse beads and interlocking bricks fabrications, similar to the one presented in the main paper but with lower resolution. We also present additional cross-stitch embroideries in Fig. 11 with various designs, such as different sizes, color palettes and styles. This demonstrates how our method can be used to fabricate diverse yet compelling embroideries.

4.2 Mosaic

We showcase the generalization capabilities of SD-πXL to other unit elements than pixel colors. As outlined in Section 4.1 of the main paper, instead of colors, our generator is capable of using any elements that can be rendered into images of equal dimensions $h \times w$, enabling the creation of images with a mosaic-like effect. Examples of this capability are demonstrated in Fig. 12, where we employ a variety of elements including colorful fruit emojis [emo 2024], black and white open icons from Font Awesome [fon 2024a] and letters in random colors from the monospace font Courier Bold [fon 2024b].

Table 3. Quantiative evaluation of pixelization methods across various metrics. We compare pixelated image abstraction (PIA) [Gerstner et al. 2012], two quantized variants of Make Your Own Sprites (MYOS) [Wu et al. 2022], VectorFusion [Jain et al. 2023], and three variants of SD-πXL: *palette*, *K-means*, and *adaptive*. The evaluation metrics are grouped into three categories: semantic similarity with CLIPScore L/14 and Human Preference Score V2 (HPSV2); fidelity to the input image with the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM); and aesthetic score with the LAION Aesthetic Predictor Scores [Schuhmann 2023]. Scores are provided for images of sizes 32×32, 48×48, and 64×64, along with their averages (Avg). MYOS does not provide non-integer scaling factor, and as input images have size 1024×1024, output images of size 48×48 are not provided. The higher the value, the better. Best scores across categories are in bold.

		Semantic								Fidelity								Aesthetics			
		CLIPScore L/14 ↑ [Radford et al. 2021]				HPSV2 ↑ [Wu et al. 2023]				PSNR ↑ [Horé and Ziou 2010]				SSIM ↑ [Wang et al. 2004]				Aesthetic Predictor ↑ [Schuhmann 2023]			
Method		32	48	64	Avg	32	48	64	Avg	32	48	64	Avg	32	48	64	Avg	32	48	64	Avg
PIA		21.1	23.5	25.4	23.3	0.240	0.248	0.254	0.247	14.1	**15.0**	15.6	14.9	0.455	0.485	0.487	0.476	4.47	4.59	4.85	4.64
MYOS	*palette*	18.6	—	21.7	21.2	0.231	—	0.239	0.235	12.4	—	13.7	13.1	0.407	—	0.418	0.412	4.38	—	5.04	4.71
	K-means	20.7	—	26.1	23.4	0.245	—	0.261	0.253	**15.2**	—	**16.9**	**16.0**	**0.467**	—	**0.511**	**0.489**	4.52	—	4.99	4.76
VectorFusion		**24.5**	**25.2**	27.0	25.6	**0.250**	0.256	0.261	**0.256**	8.0	8.2	8.3	8.2	0.293	0.292	0.258	0.281	4.16	4.59	4.86	4.54
SD-πXL	*palette*	23.1	22.6	24.5	23.4	0.237	0.247	0.244	0.243	10.3	11.7	11.3	11.1	0.388	0.444	0.398	0.410	**4.83**	4.46	5.13	4.81
	K-means	22.7	24.0	25.8	24.2	0.235	0.251	0.250	0.245	12.9	14.0	13.9	13.6	0.442	**0.486**	0.451	0.460	4.78	**4.66**	**5.31**	**4.92**
	adaptive	23.9	25.0	**28.4**	**25.8**	0.238	**0.261**	**0.263**	0.254	8.3	8.7	9.5	8.8	0.333	0.373	0.374	0.360	3.84	4.21	4.79	4.28

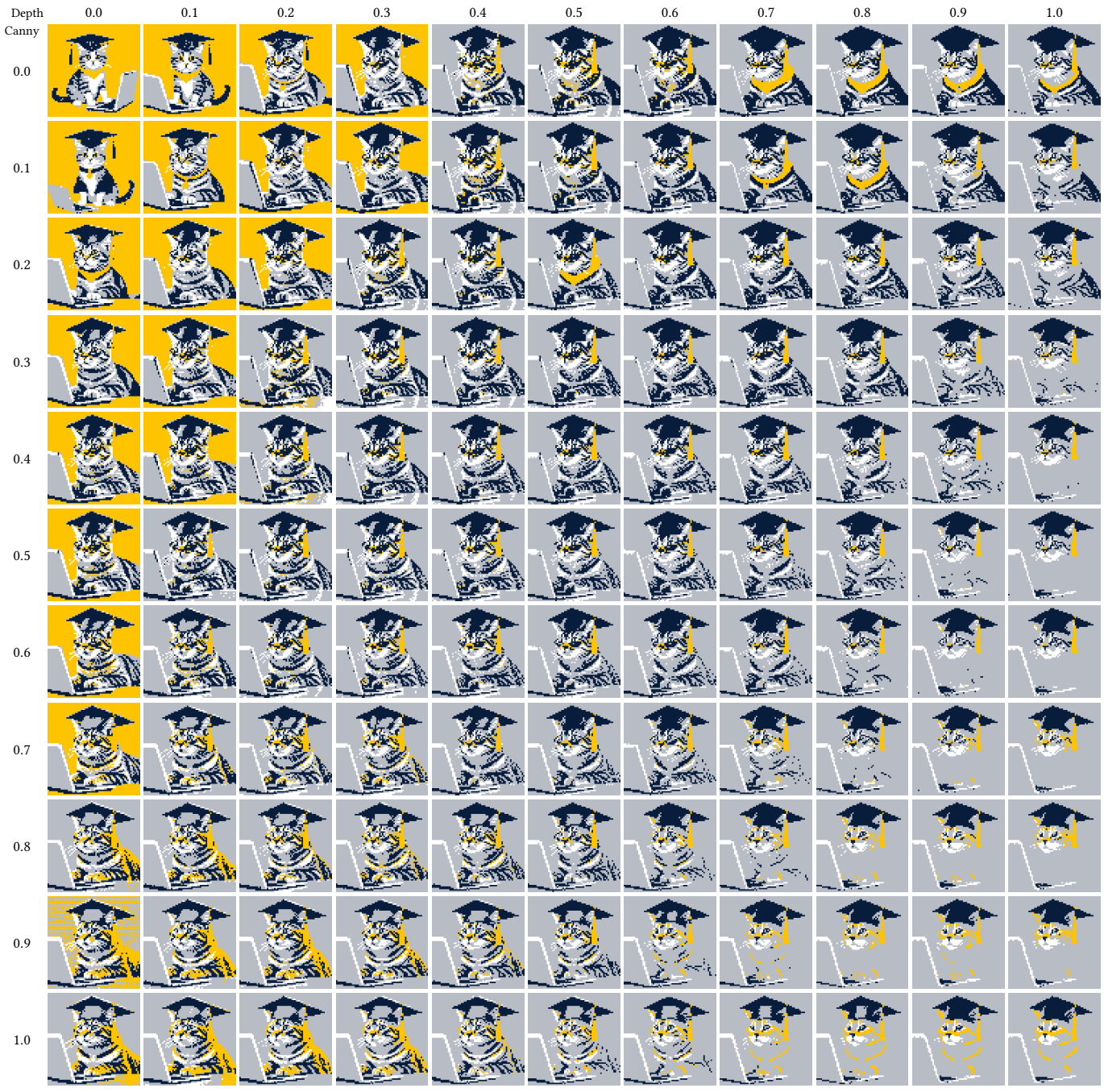


Fig. 13. We present the combined effects of ControlNet [Zhang et al. 2023] weights on both Canny edge and depth-conditioning networks [von Platen et al. 2022], examined concurrently. We initialize the weights of the generator randomly to disambiguate the contribution of ControlNet to the spatial fidelity of the generation from the influence of the initialization. The cumulative nature of these weights accounts for the distortion observed in the output when their sum exceeds 1.

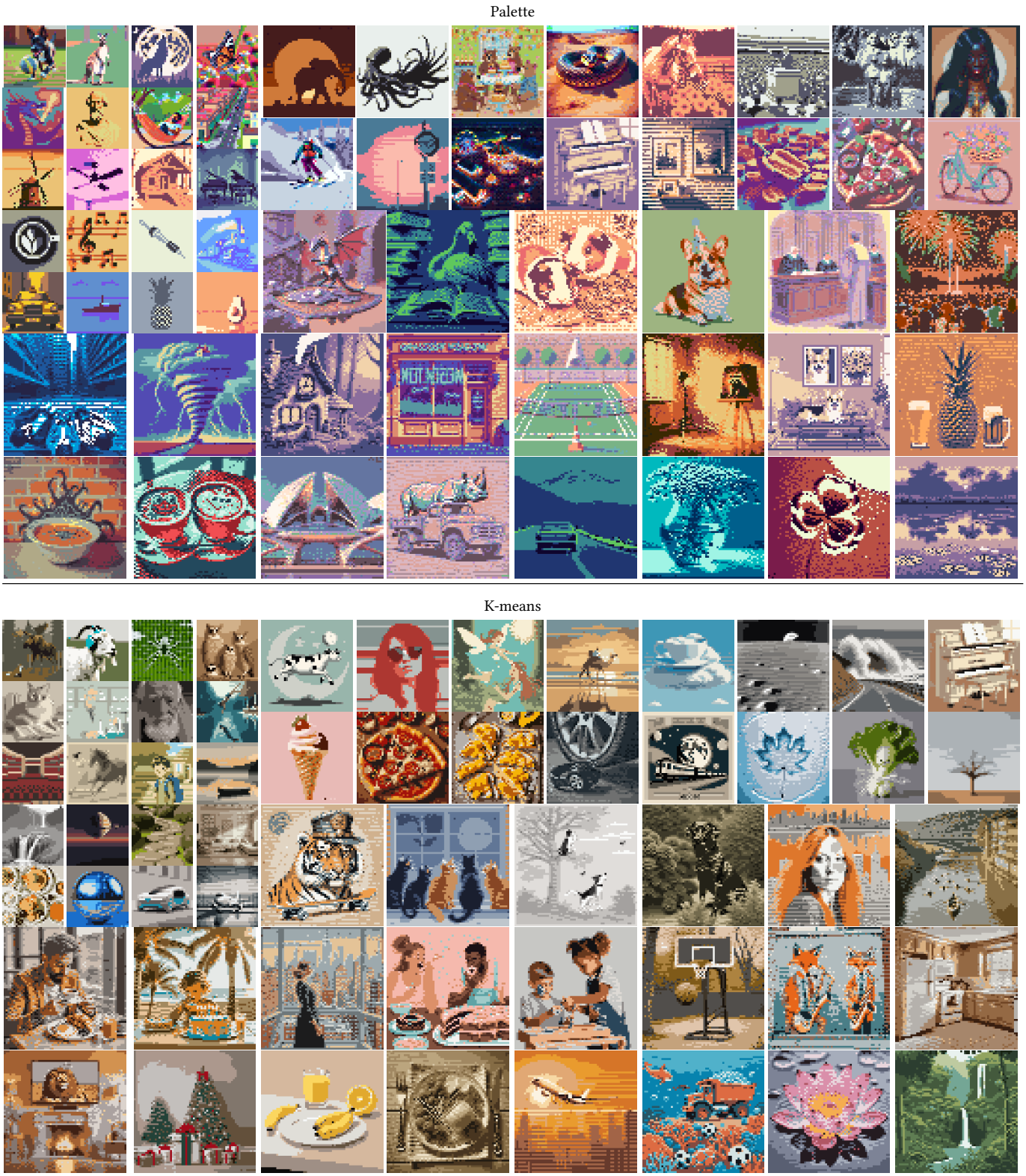
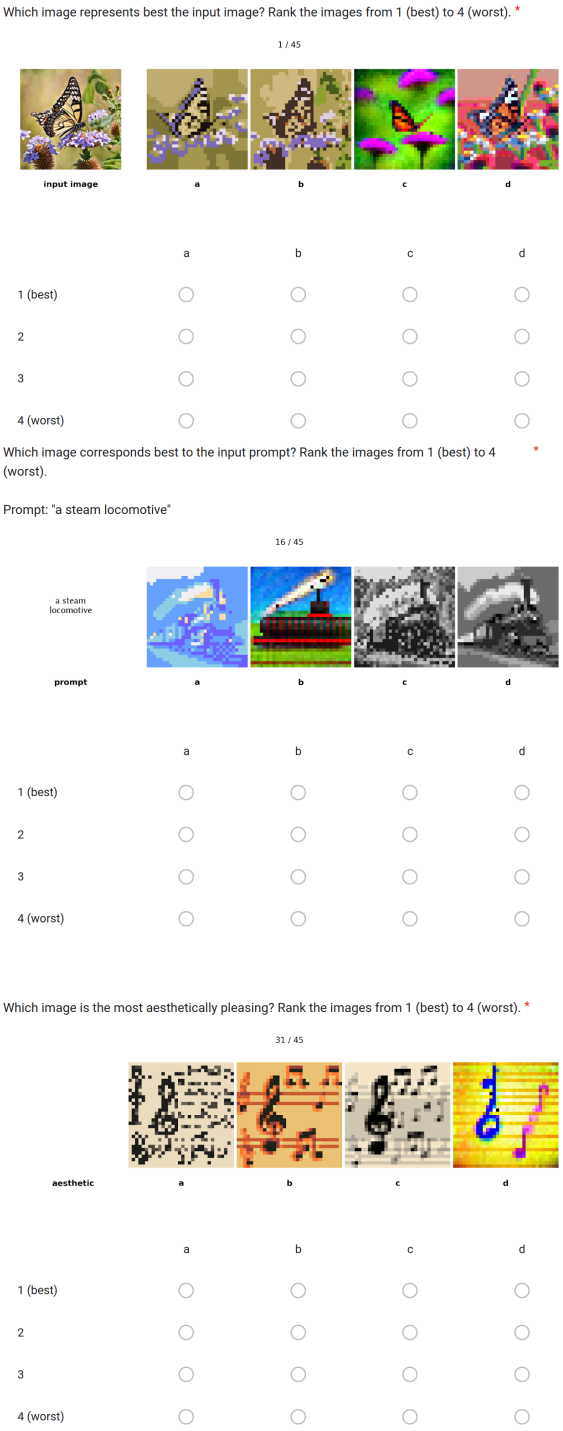


Fig. 14. Results of SD-πXL on the dataset for quantitative evaluation are presented for both *palette* (top) and *K-means* (bottom) versions. Results in three different sizes are shown: 32×32, 48×48, and 64×64, each scaled proportionally to its dimensions.

User study questionnaire

Histogram of user responses



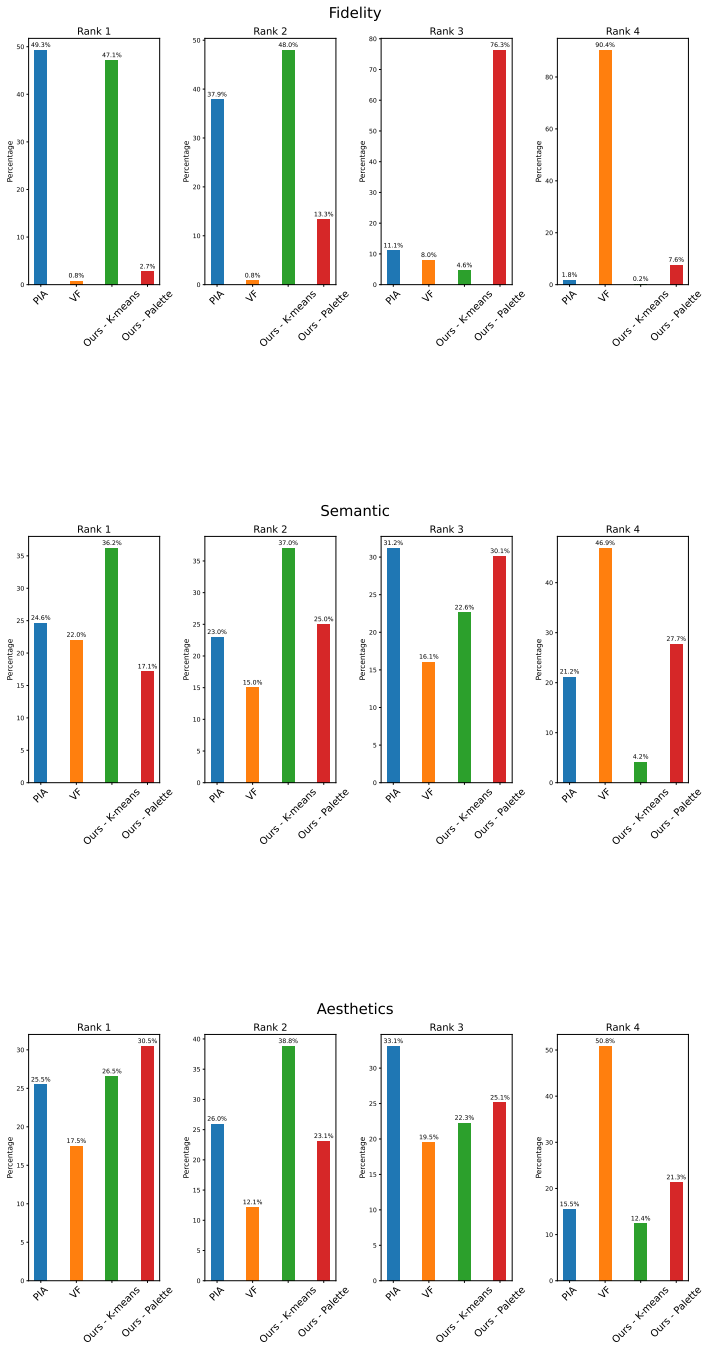


Fig. 15. Overview of the user study design and outcomes. The left column displays the questionnaires used in the study, categorized into three sections: image fidelity (first 15 questions), prompt similarity (next 15 questions), and aesthetics preference (last 15 questions). Presentation order of different methods is randomized in each section. The right column visualizes the aggregate user responses across these categories as histograms.

REFERENCES

2024. EmojiTerra. https://emojiterra.com/fruits/ Last accessed on January 21, 2024.

2024a. Font Awesome. https://fontawesome.com/ Last accessed on January 21, 2024.

2024b. Fonts Geek. https://fontsgeek.com/fonts/Courier-BOLD Last accessed on January 21, 2024.

2024. Lospec. https://lospec.com/ Last accessed on January 21, 2024.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. 2023. Improving Image Generation with Better Captions. https://cdn.openai.com/papers/dall-e-3.pdf

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

John Canny. 1986. A Computational Approach To Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-8 (12 1986), 679 – 698. https://doi.org/10.1109/TPAMI.1986.4767851

Patrick Esser, Robin Rombach, and Björn Ommer. 2021. Taming Transformers for High-Resolution Image Synthesis. arXiv:2012.09841 [cs.CV]

Timothy Gerstner, Doug DeCarlo, Marc Alexa, Adam Finkelstein, Yotam Gingold, and Andrew Nealen. 2012. Pixelated Image Abstraction. In *NPAR 2012, Proceedings of the 10th International Symposium on Non-photorealistic Animation and Rendering*.

Chu Han, Qiang Wen, Shengfeng He, Qianshu Zhu, Yinjie Tan, Guoqiang Han, and Tien-Tsin Wong. 2018. Deep Unsupervised Pixelization. *ACM Trans. Graph.* 37, 6, Article 243 (dec 2018), 11 pages. https://doi.org/10.1145/3272127.3275082

Alain Horé and Djemel Ziou. 2010. Image Quality Metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*. 2366–2369. https://doi.org/10.1109/ICPR.2010.579

Ajay Jain, Amber Xie, and Pieter Abbeel. 2023. VectorFusion: Text-to-SVG by Abstracting Pixel-Based Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1911–1920.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. arXiv:1611.01144 [stat.ML]

Kornel Lesiński. 2024. *libimagequant*. https://github.com/ImageOptim/libimagequant

Tzu-Mao Li, Michal Lukáč, Gharbi Michaël, and Jonathan Ragan-Kelley. 2020. Differentiable Vector Graphics Rasterization for Editing and Learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 39, 6 (2020), 193:1–193:15.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101 [cs.LG]

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. arXiv:1611.00712 [cs.LG]

Madebyollin. 2023. TAESDXL model. https://huggingface.co/madebyollin/taesdxl. Last accessed on January 23, 2024.

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]

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. arXiv:2307.01952 [cs.CV]

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv* (2022).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV]

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. 2021. Vision Transformers for Dense Prediction. *CoRR* abs/2103.13413 (2021). arXiv:2103.13413 https://arxiv.org/abs/2103.13413

Christoph Schuhmann. 2023. Improved Aesthetic Predictor. https://github.com/christophschuhmann/improved-aesthetic-predictor. Last accessed on January 19, 2024.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. 2022. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. 2023. Human Preference Score v2: A Solid Benchmark for Evaluating Human Preferences of Text-to-Image Synthesis. arXiv:2306.09341 [cs.CV]

Zongwei Wu, Liangyu Chai, Nanxuan Zhao, Bailin Deng, Yongtuo Liu, Qiang Wen, Junle Wang, and Shengfeng He. 2022. Make Your Own Sprites: Aliasing-Aware and Cell-Controllable Pixelization. *ACM Trans. Graph.* 41, 6, Article 193 (nov 2022), 16 pages. https://doi.org/10.1145/3550454.3555482

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. 2022. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. arXiv:2206.10789 [cs.CV]

Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. 2023. Text-to-3D with Classifier Score Distillation. arXiv:2310.19415 [cs.CV]

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding Conditional Control to Text-to-Image Diffusion Models.