# SD-$\pi$XL: Generating Low-Resolution Quantized Imagery via Score Distillation

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

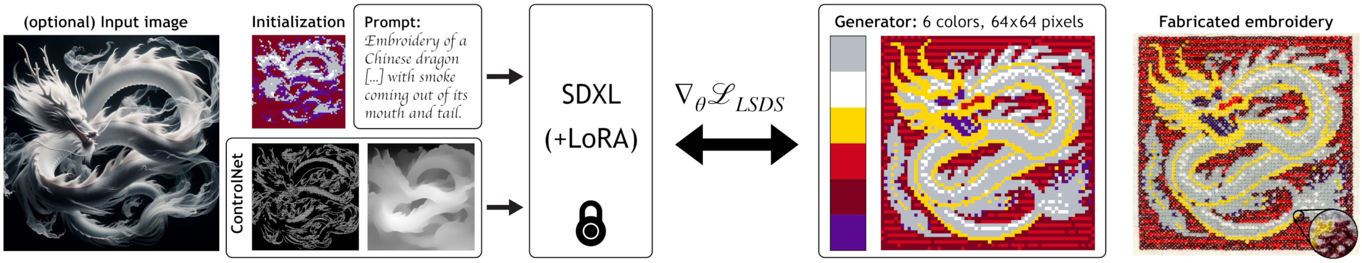
(optional) Input image
Initialization
Prompt:
Embroidery of a Chinese dragon [...] with smoke coming out of its mouth and tail.
ControlNet
SDXL
(+LoRA)
$\nabla_\theta \mathscr{L}_{LSDS}$
Generator: 6 colors, 64x64 pixels
Fabricated embroidery

Fig. 1. SD-$\pi$XL specializes in creating pixel art, characterized by its intentionally low resolution and limited color palette. Our method enables varying degrees of control: the input is a text prompt, and optionally a reference (high-resolution) image for initialization or spatial control. SD-$\pi$XL's output style can be adjusted using fine-tuned diffusion models. In this example, the full prompt reads "Embroidery of a Chinese dragon flying through the air on a dark background with smoke coming out of its mouth and tail.". The output pixel art can be used for crafted fabrications, such as the shown cross-stitch embroidery.

Low-resolution quantized imagery, such as pixel art, is seeing a revival in modern applications ranging from video game graphics to digital design and fabrication, where creativity is often bound by a limited palette of elemental units. Despite their growing popularity, the automated generation of quantized images from raw inputs remains a significant challenge, often necessitating intensive manual input. We introduce SD-$\pi$XL, an approach for producing quantized images that employs score distillation sampling in conjunction with a differentiable image generator. Our method enables users to input a prompt and optionally an image for spatial conditioning, set any desired output size $H \times W$, and choose a palette of $n$ colors or elements. Each color corresponds to a distinct class for our generator, which operates on an $H \times W \times n$ tensor. We adopt a softmax approach, computing a convex sum of elements, thus rendering the process differentiable and amenable to backpropagation. We show that employing Gumbel-softmax reparameterization allows for crisp pixel art effects. Unique to our method is the ability to transform input images into low-resolution, quantized versions while retaining their key semantic features. Our experiments validate SD-$\pi$XL's performance in creating visually pleasing and faithful representations, consistently outperforming the current state-of-the-art. Furthermore, we showcase SD-$\pi$XL's practical utility in fabrication through its applications in interlocking brick mosaic, beading and embroidery design.

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. SD-$\pi$XL: 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 INTRODUCTION

Pixel art is a common form of low-resolution, quantized images, characterized by its minimalist aesthetic and distinctive use of color. Each pixel is clearly visible, and even a single pixel modification can have a significant perceptual impact. This art style has gained widespread popularity in various applications, such as video games and contemporary artistic design. Its charm lies not only in its visual appeal but also in its historical significance, as it evokes the early days of video games, when hardware limitations necessitated the use of simple, low-dimensional representations with a restricted amount of colors. Pixel art continues to be employed in numerous indie games and artistic creations, capitalizing on its unique visual style and lower memory footprint.

As illustrated in Fig. 2, quantized images can reflect essential fabrication constraints or rationalization e.g. for embroidery [Igarashi and Igarashi 2022] or interlocking brick games [Zhou et al. 2023], where the production is constrained by a finite (usually small) amount of thread or brick colors. Creating pixel art from input images is a complex task, often requiring laborious manual effort. The challenges are compounded by the scarcity of suitable large, open datasets. Some common data augmentation techniques, such as rotation, color jitter, or blurring, may produce undesirable artifacts for pixel art style, worsening the dataset limitations. Due to its fabrication opportunities, a pixel art generation method should respect the following properties:

(1) *Hard constraints*: strict adherence to predefined constraints, such as input color palettes.
(2) *Resolution independence*: ability to produce crisp images of various resolutions without anti-aliasing.

*SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan*
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687570

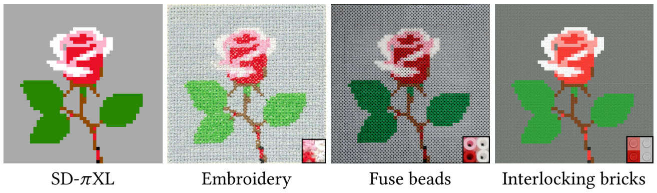

Fig. 2. SD-$\pi$XL generates low-resolution quantized images that are suitable for many fabrication applications, such as cross-stitch embroidery, fuse beads, or interlocking brick designs. The result image size is 48 × 48 pixels, generated without an initialization image, and only conditioned on the prompt "A rose flower. The branch and leaves are visible."

(3) *Flexible generation and conditioning*: ability to base generation on an input prompt or image, with adjustable semantic and geometric conditioning.
(4) *Style independence*: adaptability to different styles, such as realistic input to embroidery output as shown in Fig. 1.

As detailed in Table 1, current methods do not fully satisfy the established criteria. Existing classic and neural pixelization techniques fall short in semantic conditioning, which is crucial for pixel art to effectively communicate at low resolutions, and no method strictly adheres to specific color palettes. Fig. 3 shows limitations of current diffusion methods, as they cannot enforce strict color palette and resolution constraints, whether through prompt engineering, low-rank adaptation (LoRA) fine-tuning [Hu et al. 2021], or existing score distillation approaches [Jain et al. 2023; Poole et al. 2022].

Our paper introduces SD-$\pi$XL, a method that leverages pretrained diffusion models to generate low-resolution, quantized images within specific constraints. SD-$\pi$XL offers a versatile approach: users can input a collection of visual elements (color palettes for pixel art or sets of images for mosaics), a prompt, and optionally, an image. To create an output image of size $H \times W$ using a palette of $n$ elements, we parameterize an image generator with a tensor of dimensions $H \times W \times n$. This tensor encodes the significance of each element at every pixel position. We use Gumbel-softmax reparameterization (Sec. 3.3) to sample elements from the palette, leading to a stochastic optimization process that efficiently produces crisp pixel art while still allowing for backpropagation. We then employ diffusion networks with score distillation sampling for optimizing the parameters of the generator based on the input prompt, offering semantic understanding to the pixelization process. We also integrate spatial fidelity to the input image through conditioning on depth maps and edge detection via ControlNet [Zhang et al. 2023]. Because our approach optimizes within a predefined constraint set, adherence to the input palette is guaranteed. Our main contributions are:

(1) A differentiable image generator that strictly adheres to given constraints and works at any resolution.
(2) Evidence showing that stochastic optimization via Gumbel-softmax reparameterization produces sharp, crisp pixel art.
(3) Versatile generation capabilities from text or images, including semantic and spatial conditioning.
(4) Elimination of dataset dependency via an optimization-based method that works with any input style.
(5) State-of-the-art results in quantized image generation.

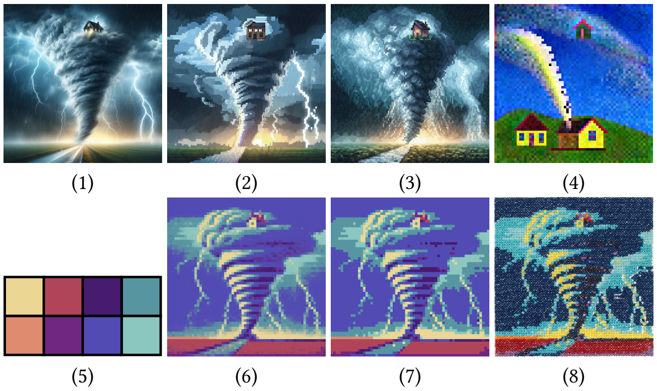

Fig. 3. Diffusion models allow for the generation of high-resolution images (1). While using a diffusion-based image translation [Podell et al. 2023; Saharia et al. 2022] with prompt-guided style is ineffective (2), fine-tuning the model for pixelized effects [Neri 2023] (3) is not generalizable across styles and requires retraining for different resolutions. VectorFusion [Jain et al. 2023] solves the resolution issue, but does not follow closely the input image (4). Our method supports outputs in any size and applies constraints to a finite palette (5), which can be enforced through either soft (6) or hard constraints (7). Color quantization further emphasizes the pixel art effect and is crucial for some fabrication applications, such as embroidery (8).

Through our experiments, we demonstrate SD-$\pi$XL's effectiveness in creating visually pleasing and accurate pixel art, surpassing existing methods. We also discuss the limitations of our approach and its potential for future work. Our supplementary material further includes ablation studies and details of our comparative evaluations. The source code is made available at https://github.com/AlexandreBinninger/SD-piXL.

## 2 RELATED WORK

In this work, we focus on generating imagery with a highly restricted number of pixels and colors. This task, requiring both semantic understanding and abstraction, is closely related to established research in color quantization and image downsampling. We review key studies in these areas to provide context and background for our approach, and refer to the survey by Kumar et al. [2019] for a comprehensive overview of the large topic of image abstraction.

*Image quantization.* Since SD-$\pi$XL utilizes image palettes with a finite discrete set of elements and operates at low spatial resolution, we discuss relevant research in the fields of color quantization and image downscaling. Content-adaptive image downscaling [Kopf et al. 2013] optimizes the shapes and locations of downsampling kernels to align with local image features, resulting in crisper output without ringing artifacts and effectively creating pixel art from vector graphics inputs. Perceptually based image downscaling [Öztireli and Gross 2015] introduces an optimization method for image downscaling that retains perceptually important features. Color manipulation is a well-studied field, often relying on layer decomposition [Aksoy et al. 2017] and manipulation via geometric tools like convex hull [Tan et al. 2016] or non-linear triads [Shugrina et al. 2020]. Colour quantization often relies on the use of a color palette. Dynamic closest color warping [Kim and Choi 2021] assesses color palette similarity by sorting and aligning colors to share a common color tendency. Floyd-Steinberg dithering [Floyd and Steinberg 1976] is an error-diffusion method that minimizes color quantization artifacts. It distributes each pixel's quantization error to adjacent pixels, creating smoother images with a defined color palette. Ozturk et al. [2014] present a brief review of color quantization and propose a method based on the artificial bee colony algorithm. Several quantization algorithms propose to abstract the input image with a non-grid clustering, to produce for instance mosaic effects [Faustino and de Figueiredo 2005], low-polygon art [Ng et al. 2018], or posterization [Chao et al. 2021]. Superpixels are groups of connected pixels that share similar characteristics, such as color or texture, forming a coherent region within an image [Ren and Malik 2003]. They can be used to segment the target image prior to clustering the color space for color quantization [Frackiewicz and Palus 2022].

*Classic pixelization methods.* Pixelated image abstraction [Gerstner et al. 2012, 2013] also relies on superpixels with a modified version of simple linear iterative clustering (SLIC) [Achanta et al. 2012] to generate pixel art-style images by simultaneously solving for feature mapping and a reduced color palette. While faithful to the input image, it lacks a semantics-aware mechanism. Automatic portrait image pixelization [Shang and Wong 2021] also relies on SLIC to introduce a pixelization algorithm for portrait images. The art-oriented pixelation (AOP) method [Lei et al. 2023] converts cartoon images into pixel art through an iterative procedure involving gridding the image, extracting its content, and separately pixelating the contour and non-contour parts of the image. Kuo et al. [2016] develop a method to animate pixel art by optimizing feature lines on each frame. Vector graphics is also present in the context of pixel-art creation. Inglis and Kaplan [2012] devise a pixelation algorithm for rasterizing vector line art while maintaining pixel art conventions. Conversely, Kopf and Lischinski [2011] address the problem of *depixelation* in generating vector representations from pixel art images by resolving pixel-scale feature ambiguities to produce smooth, connected features. This research has led to further works about pixel art depixelation via vectorization [Alberto Dominici et al. 2020; Hoshyari et al. 2018; Matusovic et al. 2023].

Table 1. Comparison of pixelization techniques. Unlike other methods, ours allows users to enforce hard constraints on resolution and palette without additional post-processing. Classical methods provide flexibility across various resolutions or scales, whereas neural methods are typically limited to a finite set of resolutions or downscaling factors. Non-diffusion deep learning methods, albeit trainable or fine-tunable for different styles, often heavily rely on their training datasets due to a lack of semantic conditioning. VectorFusion [Jain et al. 2023] also relies on score distillation (SD) [Poole et al. 2022] to optimize the parameters of a differentiable image generator, but does not constrain the image generation to an input palette.

| | | Hard constraints | Resolution independence | Semantic conditioning | Style flexibility |
|---|---|---|---|---|---|
| classic | PIA [Gerstner et al. 2012] | ✓ | ✓ | ✗ | ✓ |
| | APIP [Shang and Wong 2021] | ✗ | ✓ | ✗ | ✗ |
| | AOP [Lei et al. 2023] | ✗ | ✓ | ✗ | ✗ |
| neural | DUP [Han et al. 2018] | ✗ | finite | ✗ | dataset |
| | MYOS [Wu et al. 2022] | ✗ | finite | ✗ | dataset |
| SD | VectorFusion [Jain et al. 2023] | ✗ | ✓ | ✓ | ✓ |
| | **SD-$\pi$XL** | ✓ | ✓ | ✓ | ✓ |

*Neural pixelization methods.* Neural techniques to generate pixelized images are not new. Current neural techniques for domain transfer often use unsupervised methods like CycleGAN [Zhu et al. 2017]. These rely on generative adversarial networks (GANs) [Goodfellow et al. 2014] to transform images between different style domains. Deep unsupervised pixelization [Han et al. 2018] generates pixel art without paired training data by using several networks dedicated to different tasks, namely transforming the input image into grid-structured images, generating pixel art with sharp edges, and recovering back the original image from the pixelized result for cyclic consistency. Kuang et al. [2021] present a pixel image generation algorithm based on CycleGAN, utilizing a nested U-Net generator structure for multi-scale feature fusion, and introducing a structure combination loss to ensure the integrity of linear structures like contours in pixel images. The *Make Your Own Sprites* method [Wu et al. 2022] produces cell-controllable pixel art by using a reference pixel art for regularizing the cell structure, and disentangling the pixelization process into cell-aware and aliasing-aware stages. Jiang and Sweetser [2022] also propose a GAN-based model for pixel art generation using the YUV color encoding system.

Generating sprites is an important aspect of pixel art creation, e.g. for game assets [Karp and Swiderska-Chadaj 2021]. Rebouças Serpa and Formico Rodrigues [2019] use deep neural networks to generate pixel art sprites from line art sketches. Their work is based on Pix2Pix [Isola et al. 2017], a general method that translates an image to a different domain. Also based on Pix2Pix, GAN-based sprites generation [Coutinho and Chaimowicz 2022a] expedites the process of creating pixel art character sprite sheets by generating target side poses based on source poses. Subsequently, Coutinho and Chaimowicz [2022b] propose two modifications, namely a color palette representation and a histogram loss, and discuss the difficulties of pixel-art sprite generation using GANs. These neural methods take the stance of considering pixelization as a domain transfer problem, while we incorporate semantic conditioning for low-resolution, style-agnostic generation. This adaptability allows SD-$\pi$XL to be effective across various styles and applications.

VectorFusion [Jain et al. 2023], and concurrently to our work, SVGDreamer [Xing et al. 2024], leverage a diffusion model for semantics-aware optimization of the parameters of a differentiable vector rasterizer [Li et al. 2020] via score distillation sampling [Poole et al. 2022]. They can force the generation to a grid, producing low-resolution images, but the lack of color quantization makes their results saturated and noisy. Prior to score distillation, some methods used CLIP [Radford et al. 2021] for image abstraction, such as CLIPDraw [Frans et al. 2022] or CLIPasso [Vinker et al. 2022].

*Fabrication with quantized images.* Low-resolution and color-quantized images have various fabrication applications. Embroidery is limited by the number of thread colors. While image conversion methods exist for directionality-aware embroidery patterns [Zhenyuan et al. 2023], low-resolution pixel art is particularly adapted for cross-stitching. Though cross-stitching can be automatically performed by modern sewing machines, e.g. [PFAFF ® 2020], techniques to correct human mistakes on-the-fly for pixel art fabrication have been developed [Igarashi and Igarashi 2022]. Fuse beads is a popular form of pixel art fabrication, and is de facto limited by the available bead colors. Interlocking bricks such as LEGO® are another suitable fabrication possibility. While advancements have been made in the realm of 3D LEGO® design methodologies [Xu et al. 2019], efforts are actively made to explore the design of 2D brick-based structures as well [Zhou et al. 2023].

## 3 BACKGROUND

### 3.1 Diffusion

We briefly review diffusion models, referring the reader to a comprehensive survey for more in-depth explanation [Po et al. 2024]. Diffusion models are a family of generative models that map Gaussian noise into samples from a targeted image distribution $p_{\text{data}}$ [Ho et al. 2020; Sohl-Dickstein et al. 2015]. They consist of two main stages. The first is the forward process: an initial sample $x_0 \sim p_{\text{data}}$ undergoes a progressive noising over $T$ steps, culminating in a Gaussian-distributed sample $x_T \sim \mathcal{N}(0, \sigma_T)$. To avoid exploding variance [Song et al. 2021], the noisy sample is computed as $x_t = \alpha_t x_0 + \epsilon \sigma_t$, where $\epsilon \sim \mathcal{N}(0, 1)$, $t \in \{0, ..., T\}$ is the time step, $\alpha_t$ and $\sigma_t$ parameterize the diffusion [Kingma et al. 2023]. Following the initial phase is the backward process: it begins with a noisy sample $x_t$ and successively estimates the noise to progressively generate cleaner samples $x_{t-1}$. This iterative denoising continues until it reconstructs the final image $x_0$, which closely resembles the original data distribution $p_{\text{data}}$. Typically, this denoising function is implemented using a U-Net architecture [Ronneberger et al. 2015], denoted as $\epsilon_\phi(x_t; t)$. This function specifically aims to deduce the noise $\epsilon$ that was initially mixed with the original data $x_0$ to create the noisy version $x_t$.

*Conditioning in diffusion models.* The denoising process can be conditioned by a parameter $y$, for instance with text for prompt-based image generation. To generate samples aligned with a specific condition $y$, diffusion models utilize classifier-free guidance (CFG) [Ho and Salimans 2022]. CFG modifies the conditioned prediction $\epsilon_\phi(x_t; y, t)$ away from the unconditioned prediction $\epsilon_\phi(x_t; \emptyset, t)$, with scaling $s \in \mathbb{R}$ modulating the intensity of the conditioning:

$$\epsilon_{s,\phi}(x_t; y, t) = \epsilon_\phi(x_t; y, t) \; + \; s\,(\epsilon_\phi(x_t; y, t) - \epsilon_\phi(x_t; \emptyset, t)).$$

### 3.2 Score distillation

Score distillation employs pretrained diffusion models to compute semantics-aware gradients for updating the parameters of a differentiable renderer or generator [Poole et al. 2022]. Denote $g$ a differentiable image generator with parameters $\theta$, and $x = g(\theta)$ a generated image. For a given time step $t$, a noised version of $x$ is defined as $x_t = \alpha_t x + \epsilon \sigma_t$, with $\epsilon \sim \mathcal{N}(0, 1)$. The gradient of the score distillation sampling (SDS) loss is described by the equation

$$\nabla_\theta \mathcal{L}_{SDS} = \mathbb{E}_{t,\epsilon}\left[ w(t) \left( \epsilon_{s,\phi}(x_t; y, t) - \epsilon \right) \frac{\partial x}{\partial \theta} \right], \tag{1}$$

where $w(t) = \sigma_t^2$ serves as a scaling factor. This gradient is subsequently used to refine the parameters of the generator $g(\theta)$. Although initially developed for 3D generation, the application of score distillation extends beyond 3D. Given that an image generator is differentiable, score distillation can be used for semantics-based optimization, such as prompt-based image editing [Hertz et al. 2023]. Its utility is also evident in various other forms of image representation, such as vector graphics [Jain et al. 2023], font design [Iluz et al. 2023], or tiling [Aigerman and Groueix 2024].

### 3.3 Gumbel reparameterization

The Gumbel reparameterization technique utilizes the Gumbel distribution [Gumbel 1954] for sampling from a categorical distribution using its logits. Its impact is analyzed in Sec. 5.2, and this section explains its operation. Consider a set of $n$ scalars $(\lambda_0, ..., \lambda_{n-1})$ which represent the logits of a categorical probability distribution $Cat(\pi_0, \ldots, \pi_{n-1})$, where the probability of selecting the $k$-th category is determined by the softmax operation $\pi_k = e^{\lambda_k} / \sum_{l=0}^{n-1} e^{\lambda_l}$. Let $\{G_k\}_{0 \le k < n}$ be a series of $n$ independent random variables, each sampled from a Gumbel distribution $Gumbel(0, 1)$, and let $y_k = \lambda_k + G_k$ for $0 \le k < n$. The random variable $Y := \text{argmax}_{0 \le k < n}\{y_k\}$ is then distributed according to $Cat(\pi_0, \ldots, \pi_{n-1})$. The *Gumbel-Softmax* reparameterization technique offers a way to perform stochastic sampling from categorical distributions while remaining amenable to backpropagation [Jang et al. 2017; Maddison et al. 2017]. This method utilizes a softmax function that is parameterized by a temperature scalar $\tau$. Given $n$ categories $\{c_k\}_{0 \le k < n}$ and the objective of sampling from the categorical distribution $Cat(\pi_0, \ldots, \pi_{n-1})$, the softmax function for each category is defined as $s_k(\tau) = e^{\frac{y_k}{\tau}} / \sum_{l=0}^{n-1} e^{\frac{y_l}{\tau}}$, where $y_k$ are the logits modified by Gumbel noise. The sampling process of a category is then realized by $c_\tau = \sum_{k=0}^{n-1} s_k(\tau) c_k$. The parameter $\tau$ modulates how closely $c_\tau$ approximates a categorical distribution. As $\tau$ approaches zero, $s_k(\tau)$ converges to an indicator function $1_{k = \text{argmax}_{0 \le l < n}\{y_l\}}$, implying that for small $\tau$, $c_\tau$ closely resembles the categorical sampling $Cat(\pi_0, \ldots, \pi_{n-1})$ from the categories $\{c_k\}_{0 \le k < n}$. Conversely, as $\tau$ increases towards infinity, $s_k(\tau)$ approaches $\frac{1}{n}$, meaning larger $\tau$ values lead to $c_\tau$ resembling a uniform average of the categories.

## 4 METHOD

SD-$\pi$XL optimizes the parameters of a differentiable image generator by using SDXL [Podell et al. 2023], a pre-trained latent diffusion model, denoted as $\epsilon_\phi$, to derive a semantics-aware loss. The method requires an input text prompt $y$ and can optionally take an input image $\tilde{x}$ to guide the diffusion process. The inclusion of a smoothness loss is also supported. Our method is illustrated in Fig. 4.

### 4.1 Stochastic quantized image generation

In the proposed framework, the goal is to synthesize an image using only $n$ distinct colors from a finite set $C = \{c_k\}_{0 \le k < n}$. Although $C$ is typically a color palette—equivalent to a collection of $n$ single-pixel images—it can also represent any set of elements that are uniform in size and can be rendered as image pixels, as shown in the mosaics in the supplementary material. To generate an image $x$ of dimensions $(H, W)$ using colors from the palette $C$, we employ a generator $g$, parameterized by $\theta = \lambda_{i,j,k} \in \mathbb{R}^{H \times W \times n}$. The logits $\lambda_{i,j,k}$ give the probability that the pixel at position $(i, j)$ in $x$ will take the value $c_k$, computed as

$$\pi_{i,j,k} = \frac{e^{\lambda_{i,j,k}}}{\sum_{l=0}^{n-1} e^{\lambda_{i,j,l}}}.$$

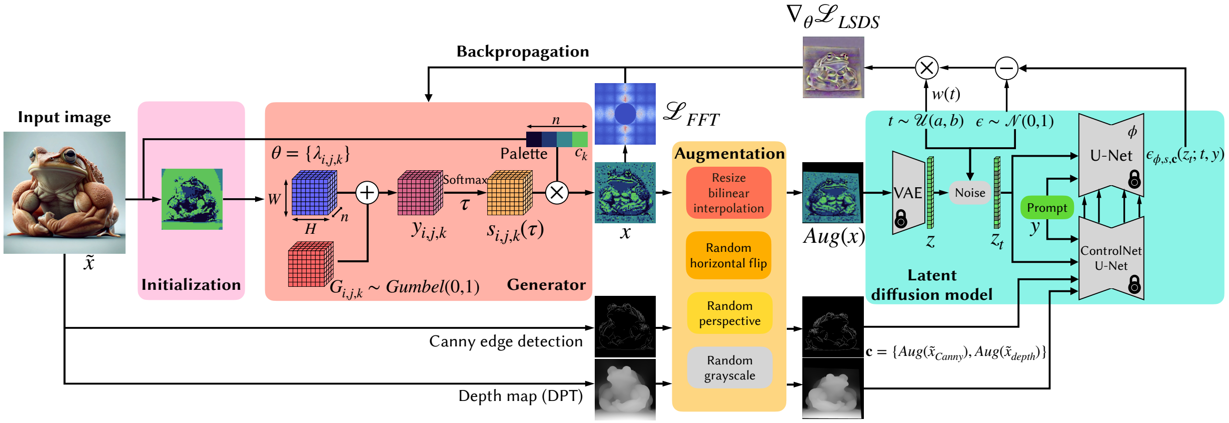

Fig. 4. Visualization of the optimization process for generating a pixelized $H \times W$ image with a color palette of size $n$. If an input image is provided, the process starts with initializing the logits $\lambda_{i,j,k}$ by downsampling the input image and matching each pixel to the nearest palette color. Otherwise, the logits are randomly initialized. Next, Gumbel-distributed random variables $G_{i,j,k}$ are added to the logits. Applying a softmax function and combining the palette colors weighted by $s_{i,j,k}(\tau)$ yields an output image $x$. This $x$, the Canny edge map [Canny 1986] and an estimated depth map [Ranftl et al. 2021] of the input image are then augmented and used in a latent diffusion model [Podell et al. 2023] to compute a semantic loss $\nabla_\theta \mathcal{L}_{LSDS}$, conditioned on an input prompt $y$. Additionally, a smoothness loss $\mathcal{L}_{FFT}$ derived from $x$ is used to optimize the parameters $\theta$.

By definition, our generator is invariant to translation of $\theta$. We take advantage of the Gumbel-softmax reparameterization (Sec. 3.3) and sample $HWn$ independent random variables $G_{i,j,k} \sim Gumbel(0,1)$, and define $y_{i,j,k} := \lambda_{i,j,k} + G_{i,j,k}$. After performing a softmax

$$s_{i,j,k}(\tau) = \frac{e^{\frac{1}{\tau} y_{i,j,k}}}{\sum_{l=0}^{n-1} e^{\frac{1}{\tau} y_{i,j,l}}},$$

the color of each pixel in $x$ is computed as $x_{i,j}(\tau) = \sum_{k=0}^{n-1} s_{i,j,k}(\tau)\, c_k$. Lower $\tau$ values enhance the resemblance of the sampling process to a categorical distribution, but excessively small $\tau$ leads to backpropagation instability. In practice, we find $\tau = 1$ to achieve reasonable results. Further insights and discussions on this choice are presented in the supplementary material.

## 4.2 Input image conditioning

SD-$\pi$XL operates with a semantic loss, yet the optimization process can be enhanced by an input image for both initialization and spatial conditioning. Since we can use rejection sampling [Jain et al. 2023] to generate images from prompt $y$ and then select the best according to their CLIP score [Radford et al. 2021], the content of this section also applies to text-only pixel art generation. We initialize the generator with an image $\tilde{x}^d$, obtained by downsampling $\tilde{x}$ to size $(H, W)$ using bilinear interpolation. We set the initial values of $\theta$ to $\lambda_{i,j,k} = -\|\tilde{x}^d_{i,j} - c_k\|$. If a color palette is not provided, we employ a K-means algorithm to partition the color space into $n$ clusters, using their centroids for the color palette $C$.

ControlNet [Chen et al. 2023] is a network architecture used to *spatially condition* the diffusion process. In our approach, we employ ControlNet networks pretrained to condition the diffusion on edges and depth information. By applying Canny edge detection [Canny 1986] and the dense prediction transformer (DPT) [Ranftl et al. 2021], we condition the diffusion process on the structural and spatial characteristics of the input image $\tilde{x}$, providing the generation with spatial context. We denote the denoiser conditioned by ControlNet on $\mathbf{c} = \{\tilde{x}_{Canny}, \tilde{x}_{depth}\}$ as $\epsilon_{\phi,\mathbf{c}}(x_t; y, t)$. The impact of ControlNet conditioning is examined in Sec. 5.3.

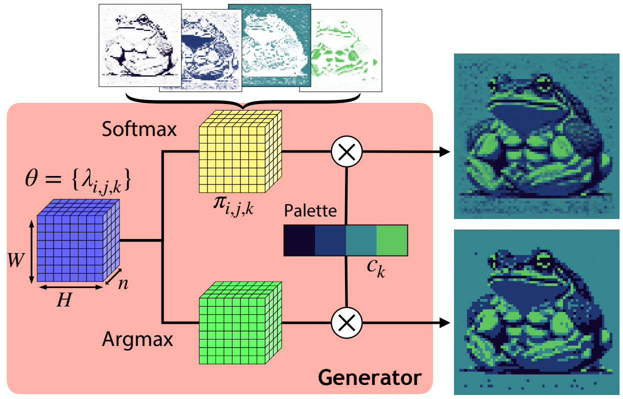

Fig. 5. Our image generator can strictly adhere to the input palette using an argmax function (bottom frog). Using softmax yields an image whose pixel colors lie in the convex hull of the input palette, leading to less crisp, pixelized outputs (top frog).

## 4.3 Image augmentation

As illustrated in Fig. 4, during our optimization, the current generated image $x$ and its associated conditioning images $\tilde{x}_{Canny}$ and $\tilde{x}_{depth}$ are fed to the diffusion model. Prior to that, we apply data augmentation: The images are first resized to the target output dimensions of the diffusion model, and subsequent augmentations include random grayscale conversion, perspective alteration, and horizontal flipping. As the conditioning images spatially guide the denoising process, it is crucial that both the generated and conditioning images undergo identical augmentations. To effectively utilize open-source latent diffusion models such as Stable Diffusion XL [Podell et al. 2023], the augmented image is encoded, represented as $z = Enc(Aug(x))$. Subsequently, we denote $z_t = \alpha_t z + \sigma_t \epsilon$ the noise-altered version of $z$ at time step $t$.

### 4.4 Loss function

Adapting the score distillation sampling loss (Eq. (1)) for latent diffusion models, the latent score distillation sampling (LSDS) loss can be written as [Jain et al. 2023]:

$$\nabla_\theta \mathcal{L}_{LSDS} = \mathbb{E}_{t,\epsilon,G}\left[w(t)\left(\epsilon_{s,\phi}(z_t; y, t) - \epsilon\right)\frac{\partial z}{\partial x}\frac{\partial x}{\partial \theta}\right]. \tag{2}$$

In our case, the expected value also takes into account the Gumbel random variables $G = \{G_{i,j,k}\}$. By decomposing $\epsilon_{s,\phi,\mathbf{c}}$, we find

$$\epsilon_{s,\phi,\mathbf{c}}(z_t; y, t) - \epsilon = \underbrace{\left(\epsilon_{\phi,\mathbf{c}}(z_t; y, t) - \epsilon\right)}_{\text{variance-reduction}} + s\underbrace{\left(\epsilon_{\phi,\mathbf{c}}(z_t; y, t) - \epsilon_{\phi,\mathbf{c}}(z_t; \emptyset, t)\right)}_{\text{semantic}}.$$

This brings a decomposition of the LSDS loss into two terms:

$$\nabla_\theta \mathcal{L}_{LSDS} = \nabla_\theta \mathcal{L}_{Noise} + s\nabla_\theta \mathcal{L}_{Sem}, \tag{3}$$

where

$$\begin{aligned} \nabla_\theta \mathcal{L}_{Noise} &= \mathbb{E}_{t,\epsilon,G}\left[w(t)\left(\epsilon_{\phi,\mathbf{c}}(z_t; y, t) - \epsilon\right)\frac{\partial z}{\partial x}\frac{\partial x}{\partial \theta}\right], \\ \nabla_\theta \mathcal{L}_{Sem} &= \mathbb{E}_{t,\epsilon,G}\left[w(t)\left(\epsilon_{\phi,\mathbf{c}}(z_t; y, t) - \epsilon_{\phi,\mathbf{c}}(z_t; \emptyset, t)\right)\frac{\partial z}{\partial x}\frac{\partial x}{\partial \theta}\right]. \end{aligned} \tag{4}$$

The noise-reduction loss component refines the parameters to yield a denoised image output, a desirable feature in contrast to its typically obstructive role in 3D generation. The semantic loss ensures that the generated result is in harmony with the provided prompt. A justification for this decomposition of the loss terms is elaborated in the supplementary material.

SD-$\pi$XL, being optimization-centric, allows for the integration of conventional loss functions. We introduce an additional fast Fourier transform (FFT) [Brigham and Morrow 1967] loss to enhance smoothness. This involves calculating the FFT of the grayscale of $x$, centering it, masking out low frequencies with $M \in \mathbb{R}^{H\times W}$, and averaging the absolute values:

$$\mathcal{L}_{FFT} = \frac{\|Shift(FFT(x)) \odot M\|_1}{\|M\|_1}. \tag{5}$$

Finally, the gradient of our loss can be written as:

$$\nabla_\theta \mathcal{L} = \nabla_\theta \mathcal{L}_{Noise} + s\nabla_\theta \mathcal{L}_{Sem} + w_{FFT}\nabla_\theta \mathcal{L}_{FFT}. \tag{6}$$

In practice, we find $s = 40$ and $w_{FFT} = 20$ to yield effective results.

## 5 RESULTS

This section outlines the final image generation process after optimization and justifies the adoption of the Gumbel-softmax reparameterization. We succinctly present the influence of ControlNet and the results of our comparative analysis, and refer the reader to the supplementary material for further details. We end with a discussion of our method's limitations and future directions.

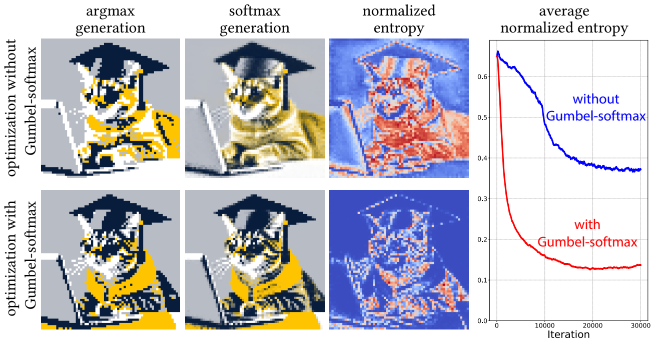

Fig. 6. We show SD-$\pi$XL's results with the Gumbel-softmax reparameterization (first row) and without (second row) during the optimization. The argmax-generation, the softmax-generation, the entropy per pixel and the average normalized entropy over time are displayed. Images are 64 × 64 pixels. The average normalized entropy is shown for 30,000 steps to ensure that the obtained results are not due to an early stop.

### 5.1 Final image generation

After optimization, our generator offers two image generation methods, shown in Fig. 5. The first option is *argmax-generated* images, which respect hard constraints and strictly adhere to a color palette,

$$x_{i,j} = c_{\tilde{k}_{i,j}}, \quad \text{where } \tilde{k}_{i,j} = \text{argmax}_{0\le k<n}\lambda_{i,j,k}.$$

The second option is using $\pi_{i,j,k}$ as coefficients of a convex sum over the palette $C$ to obtain *softmax-generated* images, calculated as

$$x_{i,j} = \textstyle\sum_{k=0}^{n-1} \pi_{i,j,k}\, c_k.$$

Their color space is merely constrained to the convex hull of the palette $C$, softening the pixel art effect. We showcase in Fig. 10 the two generation methods. Note that softmax-generated images do not require Gumbel reparameterization during optimization, as explained in the following section.

### 5.2 Stochastic vs. deterministic optimization

We explain the rationale behind including Gumbel reparameterization during optimization for argmax-generated images. In the stochastic optimization process, with Gumbel reparameterization, $\pi_{i,j,k}$ is interpreted as the likelihood of the element in position $(i, j)$ being $c_k$. Conversely, deterministic optimization (without Gumbel reparameterization) alters this perception, treating $\pi_{i,j,k}$ as coefficients in a convex combination of palette elements. This approach enables the generation of stylized low-resolution images through softmax-generation, as exemplified by the cat's fur texture in Fig. 6, but it adversely impacts the readability of argmax-generated images.

To explain this phenomenon, we analyze the entropy per pixel. The entropy of a probability distribution quantifies its uncertainty [Shannon 1948], and is defined as $H(\pi_{i,j}) := -\sum_{k=0}^{n-1} \pi_{i,j,k}\log(\pi_{i,j,k})$. Given that a uniform distribution represents the peak of categorical distribution entropy, the maximum entropy is $\log n$. Thus, we use normalized entropy $\bar{H}(\pi_{i,j}) := \frac{1}{\log n}H(\pi_{i,j})$ to gauge pixel uncertainty independently of the palette size. Our findings reveal that

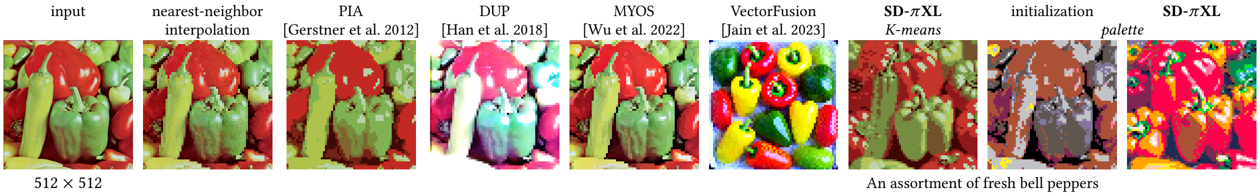
input
nearest-neighbor interpolation
PIA [Gerstner et al. 2012]
DUP [Han et al. 2018]
MYOS [Wu et al. 2022]
VectorFusion [Jain et al. 2023]
SD-πXL K-means
initialization
palette
SD-πXL
512 × 512
An assortment of fresh bell peppers

Fig. 7. Visual comparison of pixelization methods with a downscale factor of 8. The input image is displayed with its size indicated below. Both VectorFusion and SD-$\pi$XL are initialized with the input image as their initial state, and conditioned on the prompt indicated below their results. We 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-$\pi$XL operate within a 8-color limit, nearest-neighbor interpolation, DUP, MYOS, and VectorFusion have no such constraints and are not quantized.

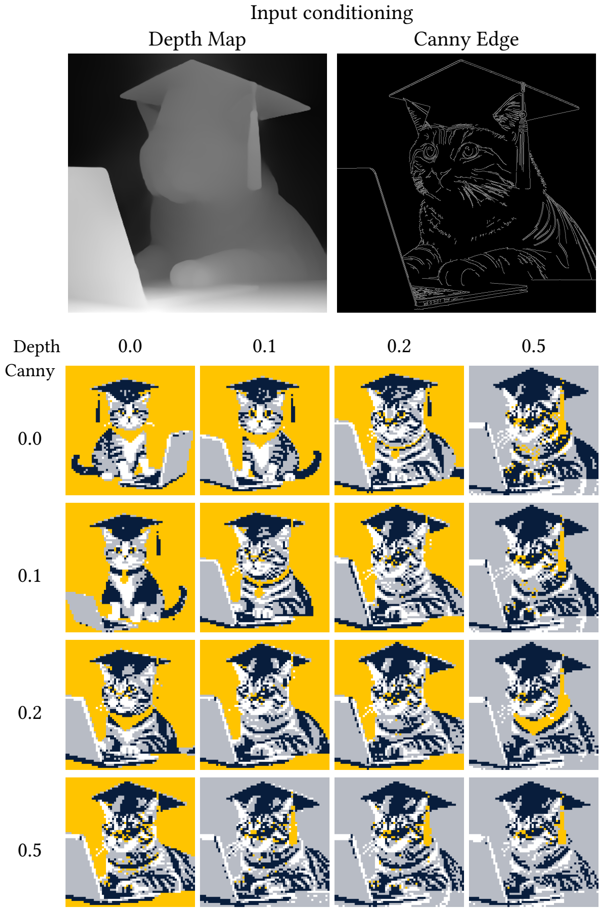

Fig. 8. We present the combined effects of ControlNet [Zhang et al. 2023] weights on both Canny edge and depth-conditioning networks [von Platen et al. 2022], examined concurrently. The image is randomly initialized to disambiguate the contribution of ControlNet from the influence of the initialization.

the Gumbel reparameterization significantly reduces entropy, as displayed on Fig. 6. Due to the pixel-wise independence in samples, employing the Gumbel-softmax reparameterization introduces noise in the results, which serves a beneficial purpose during the optimization phase, as the loss function is designed to counteract this noise. A probability distribution nearing uniformity, indicated by high entropy, leads to noisier images. Therefore, our optimization achieves denoising by encouraging the logits $\lambda_{i,j,k}$ to diverge significantly, effectively pushing the softmax towards a distinct class representation for each pixel. As a result, the optimized logits lead to a clearer, less noisy output by strongly favoring one class over the others in the softmax distribution, leading to crisper, pixelized visuals and lower entropy, demonstrated in Figs. 6 and 10.

### 5.3 ControlNet influence

As explained in Sec. 4.2, the Canny edge and depth maps of the input image can spatially condition the generation via ControlNet [Zhang et al. 2023]. The user can modulate the weights used for controlling the generation, and Fig. 8 shows that incrementing ControlNet's weights increases the fidelity of the result to the input image layout. Additional comparisons are available in the supplementary material.

### 5.4 Pixelization evaluation

We extensively evaluate the use of our method for pixelization through a quantitative comparison and a perceptual study. We compare with Pixelated Image Abstraction (PIA) [Gerstner et al. 2012], quantized Make Your Own Sprite (MYOS) [Wu et al. 2022] and VectorFusion [Jain et al. 2023]. Our method is presented in two forms: the "palette" variant utilizes a predefined palette, and the "K-means" variant computes a palette from the input image using K-means clustering. We provide one visual comparison in Fig. 7, and several additional examples are provided in the supplementary material alongside additional details and result metrics of our quantitative evaluation. For our quantitative evaluation, we generate 150 images

Table 2. Evaluation through a perceptual study, highlighting the performance of SD-$\pi$XL (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 |

and pixelize them. We analyze pixelization methods across three metrics: semantic similarity, fidelity and aesthetics. The metrics show distinct strengths: VectorFusion achieves the best semantic accuracy, while MYOS and PIA lead in fidelity. Our method excels in aesthetics due to its superior color harmony. Despite the limitations imposed by color quantization, our SD-$\pi$XL variants also deliver competitive results for both semantic accuracy and fidelity, effectively balancing these objectives and providing the most aesthetically pleasing results overall. These results are corroborated in Fig. 7: the results from nearest-neighbor, PIA and MYOS are very close to the input, but at the expense of aesthetics or clarity. DUP tends to show saturated colors, and VF diverges significantly from the input image due to lack of spatial conditioning. Our method strikes a balance between fidelity and aesthetics, even on a color palette very different from the input's colors. We also conducted a perceptual study, where 56 participants evaluated 45 images sampled randomly and rated each based on the given criteria. Results, displayed in Table 2 and Table 3, showed our *K-means* variant excelling in semantic accuracy, while PIA led in fidelity. The *palette* variant was favored for its aesthetic appeal, and VectorFusion generally received lower rankings across all categories, indicating some limitations in these aspects compared to other methods.

### 5.5 Limitations and future work

We acknowledge several limitations and future research areas for SD-$\pi$XL. While our method does not require training a network from scratch, the overall optimization process can be quite slow, requiring 1.5 hours on an Nvidia RTX4090 for 6000 steps. Additionally, the model's reliance on prompts is a limitation. Further exploration into image-only semantic conditioning [Ye et al. 2023] could potentially eliminate the need for prompts and increase fidelity. Another limitation of our method is the independent sampling for each pixel. Stochastic sampling conditioned on multiple pixels or joint probability distribution between neighboring pixels could improve the awareness of the method at a more global level, which could improve its overall quality and convergence speed. Moreover, the prospect of achieving frame-to-frame consistency in pixelized animations offers a promising direction for future extensions of this work, especially as text-to-video diffusion models continue to advance [Xing et al. 2023]. On a more general level, SD-$\pi$XL is inherently constrained by the limitations of the underlying diffusion models, including ethical concerns [Birhane et al. 2021]. With further advancements in text-to-image models and diffusion techniques, we anticipate corresponding improvements in the capabilities of SD-$\pi$XL.

Table 3. 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 |

## 6 CONCLUSION

This paper introduced SD-$\pi$XL, a method for generating low resolution, color-quantized images via semantic conditioning through diffusion-based networks. Central to our approach is the ability to strictly adhere to predefined constraints, such as input color palettes, which ensures the generation of crisp pixel art. Fig. 9 shows that our method has flexible generation capabilities, working for any desired input resolution or color palette, incorporating both semantic and image-based conditioning, and is amenable to stylization via LoRA finetuning. We demonstrate through comprehensive experiments and comparative studies the performance of SD-$\pi$XL in generating quantized images that are not only visually appealing but also accurate to the specified constraints. Our technical contribution consists in the use of the Gumbel-softmax reparameterization, justified both on the theoretical and empirical front for pixel art generation. Moreover, SD-$\pi$XL's state-of-the-art results in quantized image generation are evident in its ability to produce pixel art that meets modern-day fabrication and design requirements. Thanks to its strict adherence to a given palette, it can be directly utilized to create instructions for crafting with beads, interlocking bricks, or to embroider images using discrete styles such as cross-stitch. We produced several such physical creations, shown in Figs. 1, 2, 3. We believe that SD-$\pi$XL offers a powerful tool for artists, game developers and designers, helping make pixel art creation more accessible and versatile.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback and Danielle Luterbacher for her help with setting up the embroidery machine. Ximing Xing's open-source version of VectorFusion was instrumental in the development and design of our source code. This work was supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 101003104, ERC CoG MYCLOTH).

## REFERENCES

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 11 (2012), 2274–2282. https://doi.org/10.1109/TPAMI.2012.120

Noam Aigerman and Thibault Groueix. 2024. Generative Escher Meshes. , 11 pages. https://doi.org/10.1145/3641519.3657452

Yağiz Aksoy, Tunç Ozan Aydin, Aljoša Smolić, and Marc Pollefeys. 2017. Unmixing-Based Soft Color Segmentation for Image Manipulation. *ACM Trans. Graph.* 36, 4, Article 61c (jul 2017), 19 pages. https://doi.org/10.1145/3072959.3002176

Edoardo Alberto Dominici, Nico Schertler, Jonathan Griffin, Shayan Hoshyari, Leonid Sigal, and Alla Sheffer. 2020. PolyFit: Perception-aligned Vectorization of Raster Clip-art via Intermediate Polygonal Fitting. *ACM Transaction on Graphics* 39, 4 (2020). https://doi.org/10.1145/3386569.3392401

Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. 2021. Multimodal datasets: misogyny, pornography, and malignant stereotypes. arXiv:2110.01963 [cs.CY]

E. O. Brigham and R. E. Morrow. 1967. The fast Fourier transform. *IEEE Spectrum* 4, 12 (1967), 63–70. https://doi.org/10.1109/MSPEC.1967.5217220

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

Cheng-Kang Ted Chao, Karan Singh, and Yotam Gingold. 2021. PosterChild: Blend-Aware Artistic Posterization. *Computer Graphics Forum* 40, 4 (2021), 87–99. https://doi.org/10.1111/cgf.14343 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14343

Yang Chen, Yingwei Pan, Yehao Li, Ting Yao, and Tao Mei. 2023. Control3D: Towards Controllable Text-to-3D Generation. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*. Association for Computing Machinery, New York, NY, USA, 1148–1156. https://doi.org/10.1145/3581783.3612489

Flávio Coutinho and Luiz Chaimowicz. 2022a. Generating Pixel Art Character Sprites using GANs. arXiv:2208.06413 [cs.GR]

Flávio Coutinho and Luiz Chaimowicz. 2022b. On the Challenges of Generating Pixel Art Character Sprites Using GANs. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 18, 1 (Oct. 2022), 87–94. https://doi.org/10.1609/aiide.v18i1.21951

G.M. Faustino and L.H. de Figueiredo. 2005. Simple Adaptive Mosaic Effects. In *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)*. 315–322. https://doi.org/10.1109/SIBGRAPI.2005.46

Robert W. Floyd and Louis Steinberg. 1976. An Adaptive Algorithm for Spatial Greyscale. *Proceedings of the Society for Information Display* 17, 2 (1976), 75–77.

Mariusz Frackiewicz and Henryk Palus. 2022. Efficient Color Quantization Using Superpixels. *Sensors* 22, 16 (2022). https://doi.org/10.3390/s22166043

Kevin Frans, Lisa Soros, and Olaf Witkowski. 2022. CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders. , 5207–5218 pages. https://proceedings.neurips.cc/paper_files/paper/2022/file/21f76686538a5f06dc431efea5f475f5-Paper-Conference.pdf

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*.

Timothy Gerstner, Doug DeCarlo, Marc Alexa, Adam Finkelstein, Yotam Gingold, and Andrew Nealen. 2013. Pixelated image abstraction with integrated user constraints. *Computers & Graphics* 37, 5 (2013), 333–347. https://doi.org/10.1016/j.cag.2012.12.007

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:1406.2661 [stat.ML]

Emil Julius Gumbel. 1954. *Statistical theory of extreme values and some practical applications; a series of lectures.* U.S. Govt. Print. Office, Washington.

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

Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. 2023. Delta Denoising Score. , 2328-2337 pages.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. , 6840–6851 pages. 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]

Shayan Hoshyari, Edoardo Alberto Dominici, Alla Sheffer, Nathan Carr, Duygu Ceylan, Zhaowen Wang, and I-Chao Shen. 2018. Perception-Driven Semi-Structured Boundary Vectorization. *ACM Transaction on Graphics* 37, 4 (2018). https://doi.org/10.1145/3197517.3201312

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 [cs.CL]

Yuki Igarashi and Takeo Igarashi. 2022. Pixel Art Adaptation for Handicraft Fabrication. *Computer Graphics Forum* 41, 7 (2022), 489–494. https://doi.org/10.1111/cgf.14694

Shir Iluz, Yael Vinker, Amir Hertz, Daniel Berio, Daniel Cohen-Or, and Ariel Shamir. 2023. Word-As-Image for Semantic Typography. arXiv:2303.01818 [cs.CV]

Tiffany C. Inglis and Craig S. Kaplan. 2012. Pixelating Vector Line Art. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '12)*. Eurographics Association, Goslar, DEU, 21–28.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *CVPR* (2017).

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]

Zhouyang Jiang and Penny Sweetser. 2022. GAN-Assisted YUV Pixel Art Generation. In *AI 2021: Advances in Artificial Intelligence: 34th Australasian Joint Conference, AI 2021, Sydney, NSW, Australia, February 2–4, 2022, Proceedings* (Sydney, NSW, Australia). Springer-Verlag, Berlin, Heidelberg, 595–606. https://doi.org/10.1007/978-3-030-97546-3_48

Rafal Karp and Zaneta Swiderska-Chadaj. 2021. Automatic Generation of Graphical Game Assets Using GAN. In *2021 7th International Conference on Computer Technology Applications* (Vienna, Austria) *(ICCTA 2021)*. Association for Computing Machinery, New York, NY, USA, 7–12. https://doi.org/10.1145/3477911.3477913

Suzi Kim and Sunghee Choi. 2021. Dynamic Closest Color Warping to Sort and Compare Palettes. *ACM Transactions on Graphics (Proceedings SIGGRAPH)* 40, 4, Article 95 (2021), 15 pages. https://doi.org/10.1145/3450626.3459776

Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2023. Variational Diffusion Models. arXiv:2107.00630 [cs.LG]

Johannes Kopf and Dani Lischinski. 2011. Depixelizing Pixel Art. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)* 30, 4 (2011), 99:1 – 99:8.

Johannes Kopf, Ariel Shamir, and Pieter Peers. 2013. Content-Adaptive Image Downscaling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)* 32, 6 (2013).

Hailan Kuang, Nan Huang, Shuchang Xu, and Shunpeng Du. 2021. A Pixel image generation algorithm based on CycleGAN. In *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Vol. 4. 476–480. https://doi.org/10.1109/IMCEC51613.2021.9482118

M. P. Pavan Kumar, B. Poornima, H. S. Nagendraswamy, and C. Manjunath. 2019. A comprehensive survey on non-photorealistic rendering and benchmark developments for image abstraction and stylization. *Iran Journal of Computer Science* 2, 3 (Sept. 2019), 131–165. https://doi.org/10.1007/s42044-019-00034-1

Ming-Hsun Kuo, Yong-Liang Yang, and Hung-Kuo Chu. 2016. Feature-Aware Pixel Art Animation. *Computer Graphics Forum* (2016). https://doi.org/10.1111/cgf.13038

Peng Lei, Shuchang Xu, and Sanyuan Zhang. 2023. An art-oriented pixelation method for cartoon images. *The Visual Computer* (01 2023). https://doi.org/10.1007/s00371-022-02763-0

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.

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]

Marko Matusovic, Amal Dev Parakkat, and Elmar Eisemann. 2023. Interactive Depixelization of Pixel Art through Spring Simulation. *Computer Graphics Forum* 42, 2 (2023), 51–60. https://doi.org/10.1111/cgf.14743

Brandon Neri. 2023. pixel-art-xl. https://huggingface.co/nerijs/pixel-art-xl.

Ruisheng Ng, Lai-Kuan Wong, and John See. 2018. Pic2Geom: A Fast Rendering Algorithm for Low-Poly Geometric Art. In *Advances in Multimedia Information Processing – PCM 2017*, Bing Zeng, Qingming Huang, Abdulmotaleb El Saddik, Hongliang Li, Shuqiang Jiang, and Xiaopeng Fan (Eds.). Springer International Publishing, Cham, 368–377.

A. Cengiz Öztireli and Markus Gross. 2015. Perceptually Based Downscaling of Images. *ACM Trans. Graph.* 34, 4, Article 77 (jul 2015), 10 pages. https://doi.org/10.1145/2766891

Celal Ozturk, Emrah Hancer, and Dervis Karaboga. 2014. Color Image Quantization: A Short Review and an Application with Artificial Bee Colony Algorithm. *Informatica* 25, 3 (2014), 485–503. https://doi.org/10.15388/Informatica.2014.25

PFAFF®. 2020. *creative icon™2.* https://www.pfaff.com/globalassets/pfaff/Resources/en-US/471067426J_creative-icon-2_EN_web_LR.pdf

R. Po, W. Yifan, V. Golyanik, K. Aberman, J. T. Barron, A. Bermano, E. Chan, T. Dekel, A. Holynski, A. Kanazawa, C.K. Liu, L. Liu, B. Mildenhall, M. Nießner, B. Ommer, C. Theobalt, P. Wonka, and G. Wetzstein. 2024. State of the Art on Diffusion Models for Visual Computing. *Computer Graphics Forum* 43, 2 (2024), e15063. https://doi.org/10.1111/cgf.15063

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

Ygor Rebouças Serpa and Maria Andréia Formico Rodrigues. 2019. Towards Machine-Learning Assisted Asset Generation for Games: A Study on Pixel Art Sprite Sheets. In *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames).* 182–191. https://doi.org/10.1109/SBGames.2019.00032

Ren and Malik. 2003. Learning a classification model for segmentation. In *Proce. IEEE ICCV.* IEEE, 10–17.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. , 234–241 pages.

Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. 2022. Palette: Image-to-Image Diffusion Models. arXiv:2111.05826 [cs.CV]

Yunyi Shang and Hon-Cheng Wong. 2021. Automatic Portrait Image Pixelization. *Computers & Graphics* 95 (01 2021). https://doi.org/10.1016/j.cag.2021.01.008

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (1948), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Maria Shugrina, Amlan Kar, Sanja Fidler, and Karan Singh. 2020. Nonlinear color triads for approximation, learning and direct manipulation of color distributions. *ACM Trans. Graph.* 39, 4, Article 97 (aug 2020), 13 pages. https://doi.org/10.1145/3386569.3392461

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. , 10 pages.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. arXiv:2011.13456 [cs.LG]

Jianchao Tan, Jyh-Ming Lien, and Yotam Gingold. 2016. Decomposing Images into Layers via RGB-Space Geometry. *ACM Trans. Graph.* 36, 1, Article 7 (nov 2016), 14 pages. https://doi.org/10.1145/2988229

Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. 2022. CLIPasso: Semantically-Aware Object Sketching. *ACM Trans. Graph.* 41, 4, Article 86 (jul 2022), 11 pages. https://doi.org/10.1145/3528223.3530068

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.

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

Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang, Dong Xu, and Qian Yu. 2024. SVGDreamer: Text Guided SVG Generation with Diffusion Model. , 4546-4555 pages.

Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. 2023. A Survey on Video Diffusion Models. *arXiv preprint arXiv:2310.10647* (2023).

Hao Xu, Ka-Hei Hui, Chi-Wing Fu, and Hao Zhang. 2019. Computational LEGO technic design. *ACM Transactions on Graphics* 38, 6 (Dec. 2019), 1. https://doi.org/10.1145/3355089.3356504

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. 2023. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. (2023).

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

Liu Zhenyuan, Michal Piovarči, Christian Hafner, Raphaël Charrondière, and Bernd Bickel. 2023. Directionality-Aware Design of Embroidery Patterns. *Computer Graphics Forum* 42, 2 (2023), 397–409. https://doi.org/10.1111/cgf.14770

Mingjun Zhou, Jiahao Ge, Hao Xu, and Chi-Wing Fu. 2023. Computational Design of LEGO® Sketch Art. *ACM Trans. Graph.* 42, 6, Article 201 (dec 2023), 15 pages. https://doi.org/10.1145/3618306

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on.*

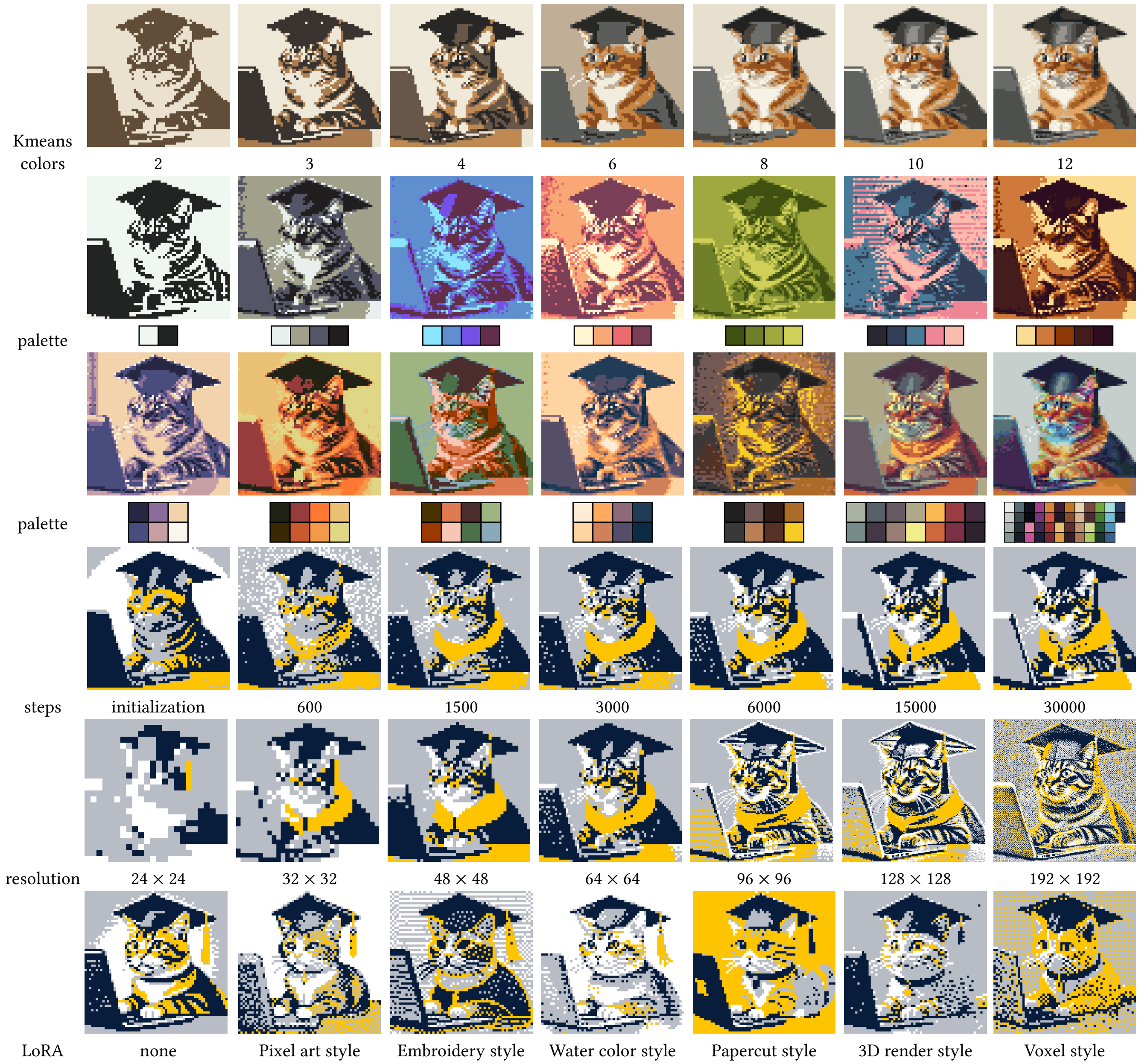

Fig. 9. The first row illustrates SD-$\pi$XL using K-means color clustering with varying numbers of colors. Rows 2 and 3 display the application of our method with different color palettes, and shows that our method works with any number of colors. The progression of SD-$\pi$XL through various time steps is depicted in row 4. In row 5, we showcase outputs at different resolutions. The final row showcases SD-$\pi$XL with diffusion models fine-tuned to distinct styles via low-rank adaption (LoRA) [Hu et al. 2021], to demonstrate the generalizability of our approach. Each name is a clickable link that directs to the corresponding LoRA. For a clearer distinction in style variations, we opt not to use ControlNet for the images in the last row. The chosen prompt for this demonstration is "A cat wearing a graduation hat using a computer", with the input image and further conditioning details provided in supplementary material.

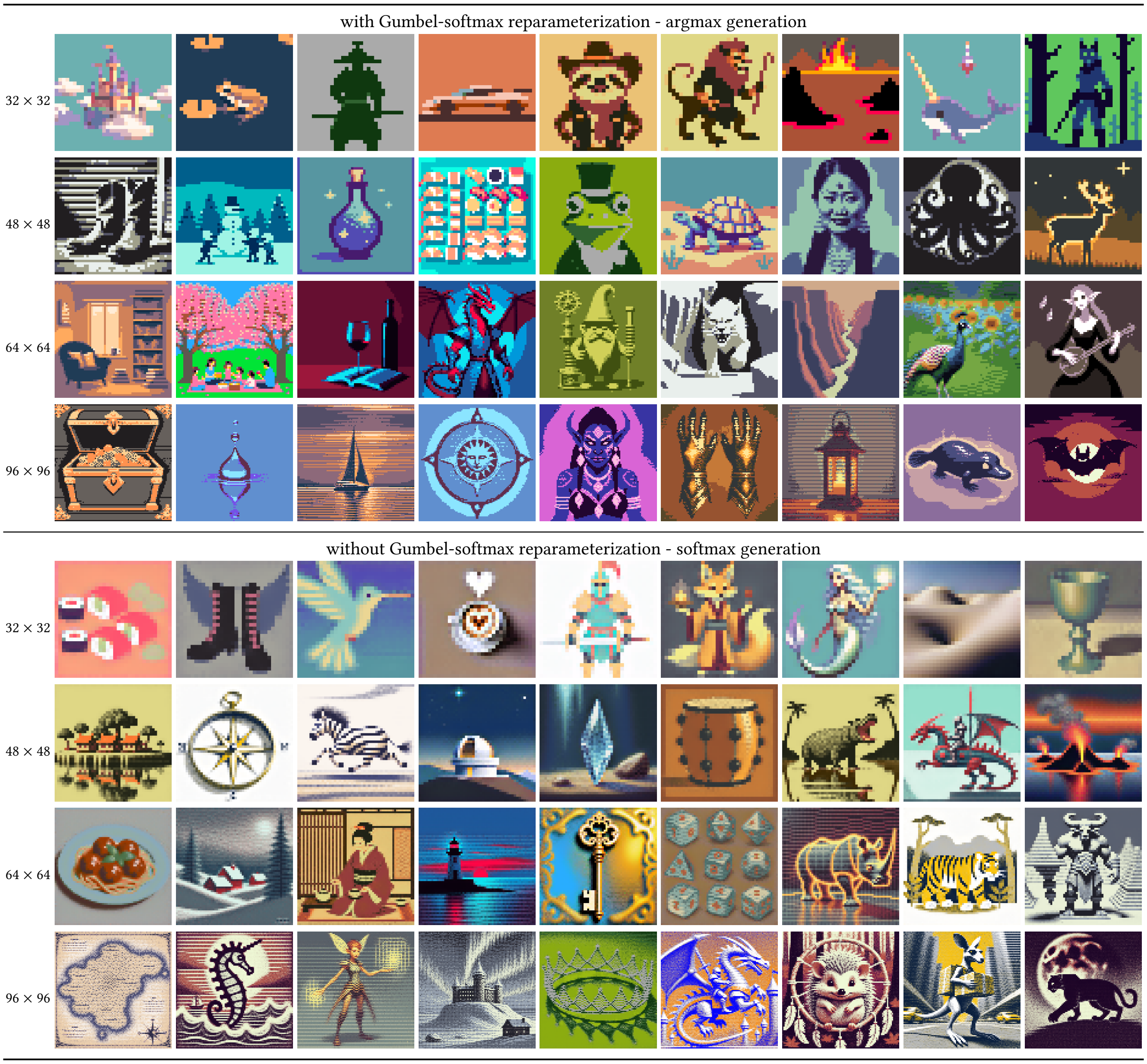
with Gumbel-softmax reparameterization - argmax generation
32 × 32
48 × 48
64 × 64
96 × 96
without Gumbel-softmax reparameterization - softmax generation
32 × 32
48 × 48
64 × 64
96 × 96

Fig. 10. Pixel art generation with SD-$\pi$XL, used without initial image or spatial conditioning. We present results on several resolutions, written on the leftmost column. The table is divided in two generation methods: the first part presents results with Gumbel-softmax reparameterization during optimization, generated with argmax. This generation method produces crisp pixel art that strictly adheres to the input palette. The second part does not use the Gumbel-softmax reparameterization, but uses a softmax generation to produce smooth, low-resolution images whose colors lie in the convex hull of the input palette.