Hyperstroke: A Novel High-quality Stroke Representation for Assistive Artistic Drawing

Haoyun Qin
University of Pennsylvania
Philadelphia, PA, USA
qhy@seas.upenn.edu

Jian Lin
Saint Francis University
Hong Kong, China
jlin@sfu.edu.hk

Hanyuan Liu
City University of Hong Kong
Hong Kong, China
hy.liu@cityu.edu.hk

Xueting Liu
Saint Francis University
Hong Kong, China
tliu@sfu.edu.hk

Chengze Li
Saint Francis University
Hong Kong, China
czli@sfu.edu.hk

Abstract

Assistive drawing aims to facilitate the creative process by providing intelligent guidance to artists. Existing solutions often fail to effectively model intricate stroke details or adequately address the temporal aspects of drawing. We introduce hyperstroke, a novel stroke representation designed to capture precise fine stroke details, including RGB appearance and alpha-channel opacity. Using a Vector Quantization approach, hyperstroke learns compact tokenized representations of strokes from real-life drawing videos of artistic drawing. With hyperstroke, we propose to model assistive drawing via a transformer-based architecture, to enable intuitive and user-friendly drawing applications, which are experimented in our exploratory evaluation.

CCS Concepts

• **Computing methodologies** → **Image processing**; **Shape analysis**; **Image manipulation**.

Keywords

assistive drawing, vector quantize, generative model, tokenization

ACM Reference Format:
Haoyun Qin, Jian Lin, Hanyuan Liu, Xueting Liu, and Chengze Li. 2024. Hyperstroke: A Novel High-quality Stroke Representation for Assistive Artistic Drawing. In *SIGGRAPH Asia 2024 Technical Communications (SA Technical Communications '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3681758.3697985

1 Introduction

Drawing is inherently an incremental process where artworks are created stroke-by-stroke, reflecting underlying drawing intent and locality. In this work, we investigate the problem of incremental drawing from the perspective of a drawing assistant. Our goal is to provide essential guidance to users in applying proper drawing strokes to complete visually pleasing artworks, considering the current unfinished canvas composition and the full or partial history of user strokes. Such an application enhances our understanding of the creative process and seamlessly integrates into existing artistic workflows in a suggest-then-accept manner.

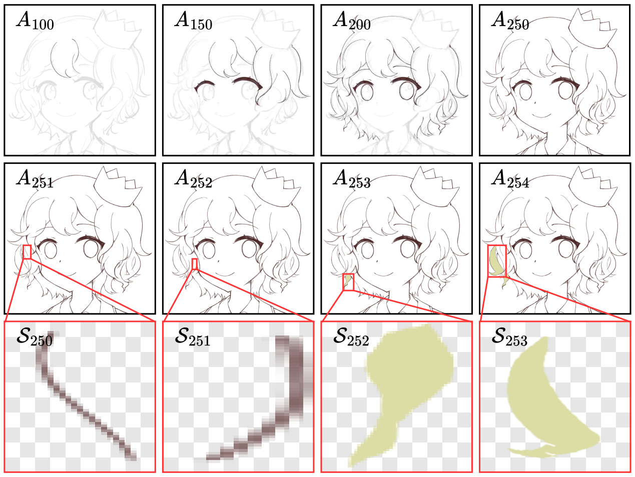

Figure 1: Example of real-life artistic drawing. Incremental drawing on canvas A_t is recorded in the form of timelapse video. The user-provided stroke $\mathcal{S}_t$ is not included in the timelapse and has to be explicitly estimated. © Linda Wei.

The existing literature focuses mainly on reproducing complete artworks using pre-defined stroke patterns [Liu et al. 2021; Singh et al. 2021; Zheng et al. 2018] or performing incremental stroke prediction exclusively in the vector domain [Bhunia et al. 2020; Ha and Eck 2017]. Recent diffusion-based models exhibit impressive results in the generation of artwork, but their generation must be performed in a single pass [Nitzan et al. 2024; Rombach et al. 2022]. This hinders iterative refinement and co-creation, which are essential in the drawing process. We hypothesize that existing approaches may prioritize overall visuals but neglect the importance of strokes, which are the fundamental basic units contributing to an artwork in both spatial and temporal domains. This oversight is particularly detrimental for a drawing assistant. In Figure 1, we illustrate several steps in which the user applies strokes. Real-life

SA Technical Communications '24, December 3–6, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1140-4/24/12
https://doi.org/10.1145/3681758.3697985

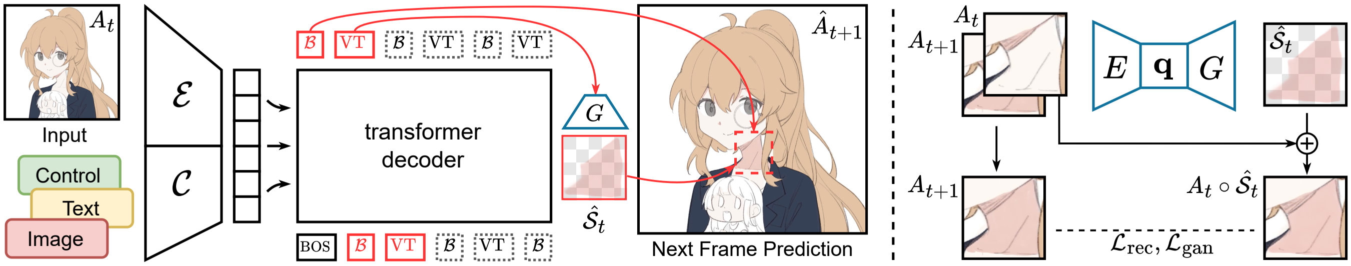
A_t
Input
Control
Text
Image
$\mathcal{E}$
$\mathcal{C}$
$\mathcal{B}$ VT $\mathcal{B}$ VT $\mathcal{B}$ VT
transformer decoder
BOS $\mathcal{B}$ VT $\mathcal{B}$ VT $\mathcal{B}$
G
$\hat{\mathcal{S}}_t$
$\hat{A}_{t+1}$
Next Frame Prediction
A_t
A_{t+1}
E q G
$\hat{\mathcal{S}}_t$
A_{t+1}
$A_t \circ \hat{\mathcal{S}}_t$
$\mathcal{L}_{\text{rec}}, \mathcal{L}_{\text{gan}}$

Figure 2: Overview of our framework. The right demonstrates the learning of tokenization in hyperstrokes (Section 2.1), while the left shows our systematic design in predictive incremental drawing (Section 2.2). Artwork involved © Linda Wei.

drawing of strokes is far more complex than simple shape primitives, involving specific movements, shape and color variations, etc. More importantly, these strokes exhibit opacity, i.e. *alpha*, to blend additively over the canvas, crafting delicate details and shadings. Therefore, understanding and modeling strokes are crucial for modeling a drawing assistant.

In this work, we propose *hyperstroke*, an efficient and expressive stroke representation to better model strokes in real-life artistic drawing with alpha-channel opacity. Our key insight is to employ a VQ-based model to learn a compact tokenized representation of grounded strokes within their bounding boxes. Our experiments demonstrate the efficiency of the hyperstroke design and, more importantly, show the potential to learn predictive incremental drawing under the hyperstroke formulation, using an encoder-decoder transformer architecture. We summarize our contributions as follows:

- We introduce a novel representation, *hyperstroke*, to model delicate artistic drawing stroke appearance and opacity;
- We propose an updated VQGAN architecture to learn hyperstroke tokenization from real-life incremental drawing;
- We investigate to use transformer models to learn hyperstroke sequences for assistive incremental drawing.

2 Method

2.1 Hyperstroke

2.1.1 Formulation. In this work, we introduce the novel *hyperstroke* representation for modelling the strokes in practical artistic drawing. Unlike traditional methods that represent storkes as simple elliptical pixels, or vector primitives, our approach aims to capture the essence of real-life strokes with diverse appearances and alpha variations. By investigating the artistic drawing process, we observe several key properties within a stroke:

- **Property 1: Independence in Representation.** Strokes are additive in nature, meaning each new stroke is an additional layer alpha-blended onto the existing canvas, independent of all other strokes, as seen in the strokes $\mathcal{S}$ of Figure 1.
- **Property 2: Spatial Sparsity.** Strokes are inherently spatially sparse. Though the canvas may be extensive, each stroke is either detailed and confined to a small area or spans a larger region but is relatively coarse. Therefore, when extracted and normalized to a consistent scale, each stroke should carry a similar amount of low-scale information.

Based on these assumptions, we design our hyperstroke representation to be atomic and compact. Leveraging the sparsity property of strokes, we propose using bounding boxes to locate each stroke and encode only the pixels within them, for better expressiveness of strokes in smaller regions. Formally, we define the pixel-domain *hyperstroke* $\mathcal{S} = \langle \mathcal{I}, \mathcal{B} \rangle$, where $\mathcal{I} = (I, \alpha)$ is a 4-channel alpha image and $\mathcal{B} = (x_1, y_1, x_2, y_2)$ is the bounding box of $\mathcal{I}$. In this way, we can regard each stroke-box combo shown in the bottom two rows of Figure 1 as a hyperstroke. When a hyperstroke $\mathcal{S}$ is applied to an image A, we denote the blending operation $A \circ \mathcal{S}$ as:

$$(A \circ \mathcal{S})(x, y) = \begin{cases} (I \cdot \alpha + A \cdot (1 - \alpha))\,(x, y) & \begin{matrix} x_1 \le x < x_2 \\ y_1 \le y < y_2 \end{matrix} \\ A(x, y). & \text{otherwise} \end{cases} \tag{1}$$

2.1.2 Tokenization. To this point, we have formulated the hyperstrokes in the pixel space. However, this formulation proves ineffective for modeling incremental drawing, as learning pixel-domain hyperstrokes with temporal information is computationally intensive. Conversely, transformer models excel at modeling temporal sequences, which is more suitable for learning incremental drawing, suggesting the representation of hyperstrokes as discrete tokens. Specifically, we perform hyperstroke tokenization separately for $\mathcal{I}$ and $\mathcal{B}$. To tokenize bounding box $\mathcal{B}$, we first subdivide the image canvas into grids of $C \times C$, with each grid cell having dimensions $\lfloor W/C \rfloor \times \lfloor H/C \rfloor$, where W and H denotes width and height of the original canvas. For any bounding box $\mathcal{B}$, we compute its smallest exterior box that snaps to the grid and tokenize it in the form $\tilde{\mathcal{B}} = (X_1, Y_1, X_2, Y_2) \in T_{\mathcal{B}}^4$, where the integer X_1, X_1, Y_2, Y_2 represents the indices of the grid corners to which the exterior box snaps, and $T_{\mathcal{B}}$ is a vocabulary of $\{0, 1, \ldots, C\}$. This grid-based design reduces the complexity of bounding box tokens without significantly compromising the encoding of the stroke image $\mathcal{I}$, using a slightly larger bounding box.

For the stroke pixels $\mathcal{I}$, we perform the same grid snapping strategy as $\mathcal{B}$, and then resize it to a consistent dimension $W_T \times H_T$. We learn to tokenize its visual tokens $\tilde{\mathcal{I}} \in T_{\mathrm{VT}}^k$ via a VQ-based approach, which will be explained in the following subsection.

2.1.3 Training Hyperstroke from Real-life Incremental Drawing. Tokenizing a 4-channel alpha image $\mathcal{I}$ appears straightforward due to existing standards such as VQGAN [Esser et al. 2021]. However, we find the quality of the data contributing to visual token learning critical. Synthesizing arbitrary alpha strokes programmatically is

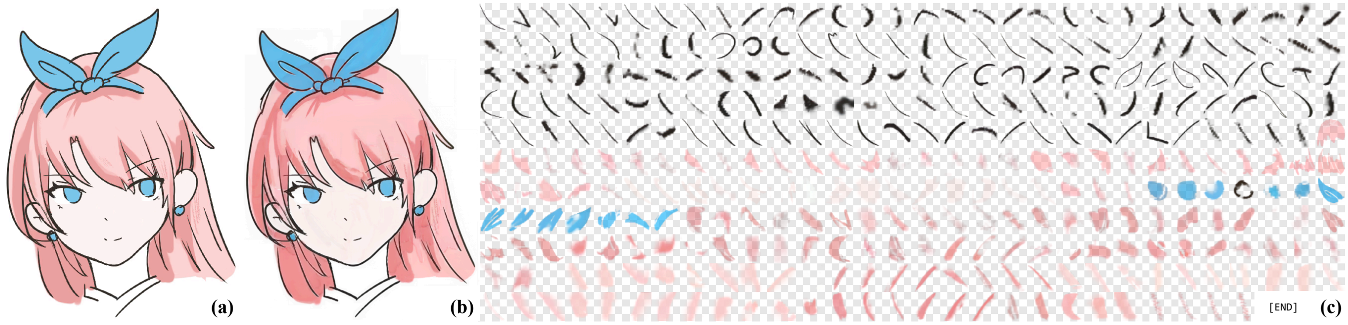

Figure 3: Reconstruction of real-life incremental drawing from timelapse videos. (a) Timelapse snapshot at $t = 328$; (b) Reconstructed canvas composited by hyperstrokes; (c) Inferred stroke sequences from adjacent timelapse frames. © Hao Chen

one direction but would overcomplicate the final encoded tokens. Real-life strokes exhibit more specific distributions, as the drawing of each stroke follows human-specific aesthetic considerations. In this circumstance, sources recording practical human-drawn strokes with pixel-level opacity would be ideal for training, but such data is usually unavailable. Therefore, we attempt to collect strokes with alpha information from *timelapse videos* (shown in Fig. 1), which capture consecutive canvas frames whenever a new stroke is applied. Unfortunately, timelapse videos do not store any specific stroke information, so we have to estimate the strokes $\mathcal{S}_t$ from adjacent frame correspondences; but direct estimation is infeasible due to the ill-posed nature of inversing alpha blending. To address this, we propose an improved VQ model architecture to predict alpha strokes $\hat{\mathcal{S}}_t$ from adjacent frames with implicit supervision, without requiring ground truth stroke.

We illustrate our VQ model design on the right of Figure 2. The input is the concatenation ($[A_t, A_{t+1}] \in \mathbb{R}^{H\times W\times 6}$) of any adjacent frames A_t and A_{t+1} in the data set. We use the encoder E and a codebook $\mathcal{Z}$ to learn the tokenization of stroke features as $\mathbf{q}(E(A_t, A_{t+1}))$. We use a decoder G to learn the reconstruction of the 4-channel stroke $\hat{\mathcal{S}}_t$ from the learned tokens. Here, we supervise $\hat{\mathcal{S}}_t$ by checking if A_{t+1} can be obtained by blending A_t with $\hat{\mathcal{S}}_t$:

$$\begin{aligned}\mathcal{L}_{\text{VQ}} = \mathcal{L}_{\text{rec}}\left(A_{t+1}, \left(A_t \circ \hat{\mathcal{S}}_t\right)\right) &+ \left\|\text{sg}\left[E(A_t, A_{t+1})\right] - z_{\mathbf{q}}\right\|_2^2 \\ &+ \left\|\text{sg}\left[z_{\mathbf{q}}\right] - E(A_t, A_{t+1})\right\|_2^2,\end{aligned} \tag{2}$$

where $\mathcal{L}_{\text{rec}}$ is the sum of the MSE loss and the perceptual loss [Zhang et al. 2018] and the other two loss terms optimize the use of codebooks; sg$[\cdot]$ denotes the stop-gradient operation. The blending operation $A_t \circ \hat{\mathcal{S}}_t$ in the supervision encourages the encoder E to focus on a decoupled representation of the stroke, rather than memorizing A_t and A_{t+1}. Besides, we also introduce adversarial learning with a discriminator D for better decoder perceptual quality, with a similar implicit supervision:

$$\mathcal{L}_{\text{GAN}}(\{E, G, \mathcal{Z}\}, D) = \left[\log D(A_{t+1}) + \log\left(1 - D\left(A_t \circ \hat{\mathcal{S}}_t\right)\right)\right]. \tag{3}$$

2.1.4 Data and Training Details. We construct our dataset in two parts: a synthetic dataset and data from real-life timelapse videos. For the synthetic data, we first perform a random crop of real artistic drawings. After that, we synthesize a Bezier stroke with varying widths and opacity and blend it with the cropped drawing to form the data. Since the synthetic data contains ground truth alpha for the stroke, we can use direct reconstruction loss $\mathcal{L}_{\text{rec}}$ in Eq. 2 instead of implicit supervision with additional alpha blending on the generator output. This direct supervision helps the model better understand opacity from the very beginning of training, thereby improving its learning capability on real-life data. Overall, our dataset consists of 85,425 synthetic data samples and 74,286 real data samples in the form of frame pairs. We mix the two types of supervision during training directly.

2.2 Learning Drawing with Hyperstroke

Expanding on the stroke tokenization method outlined in Section 2.1, we define incremental drawing as a sequence generation task, which can be effectively modeled with an encoder-decoder transformer model. The model, as shown on the left of Figure 2, leverages the encoder $\mathcal{E}$, a Vision Transformer (ViT) [Dosovitskiy et al. 2020], to extract contextual information τ_c from the current canvas A_t. Furthermore, we use the CLIP model [Radford et al. 2021] C to encode the guidance τ_g of controlling signals such as reference images and text descriptions. We combine τ_c and τ_g embeddings and send them to the decoder through cross-attention, to predict subsequent *hyperstroke* tokens $\left((\tilde{\mathcal{B}}, \tilde{\mathcal{I}}) \in T_{\mathcal{B}}^4 \times T_{\text{VT}}^k\right)^n$ in an autoregressive manner, where k is the number of visual tokens for each stroke, and n is the number of *hyperstrokes* to be predicted. With the VQ decoder model G, we will be able to decode and composite each hyperstroke back into the pixel domain, to form future frames from A_{t+1} to A_{t+n}.

We choose an encoder-decoder architecture over a decoder-only model to meet the unique needs of drawing tasks. Compared with text sequences where self-attention effectively captures context, predictive drawing involves more complex contextual requirements. The focus within is to determine the next few strokes, in the context of the current canvas composition and a few past user strokes. This complexity makes a decoder-only architecture impractical, as relying on a long sequence of historical hyperstrokes would be computationally inefficient. Conversely, our encoder model $\mathcal{E}$ directly provides the current canvas context through a Vision Transformer, eliminating the need to learn indirectly from the complete historical hyperstroke sequences. This approach provides several practical applications with the context provided, including: (a) unconditional

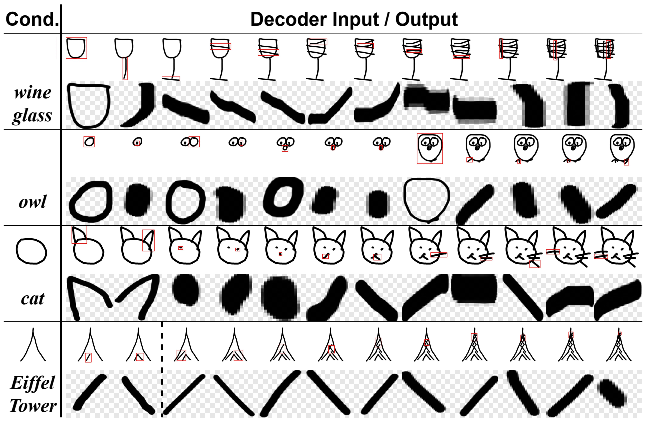

Figure 4: Results on predictive incremental drawing conditioned on raster canvas and text descriptions. Odd rows show predicted compositions; even rows demonstrate decoded grounded strokes within its bounding box. The last example prompts 2 hyperstrokes in the decoder.

sequential hyperstroke prediction; (b) prediction of subsequent hyperstrokes using a few hyperstrokes as historical prompts, for temporal-consistent stroke prediction; and (c) predicting the next visual token $\tilde{\mathcal{I}}$ given a bounding box prompt. One might argue that making the decoder output a single hyperstroke would suffice, as the rasterized next-frame context could be rendered on-the-fly. However, this method fails to capture temporal information. Our approach, by predicting ordered stroke sequences, inherently captures locality of neighboring strokes, semantics of different canvas areas, as well as the drawing intent of the artists, enabling long-term understanding capability, and thus bringing better interactivity for the artists. During training, notice that the amount of generated visual tokens $\tilde{\mathcal{I}}$ and bounding box tokens $\tilde{\mathcal{B}}$ are unbalanced, to stabilize the training, we further impose a coefficient $\lambda = k/4$ on the parts of the cross entropy loss corresponding to the generated bounding box tokens.

3 Experiments

Hyperstroke Representation. We first investigate the expressiveness of the hyperstroke representation. We use our revamped VQGAN model described in Sec. 2.1 to reconstruct all intermediate strokes from a complete artistic drawing timelapse of 328 frames (Fig. 3 (a)). Figure 3 (c) shows the reconstruction of strokes, grounded by their bounding box areas. The results demonstrate that tokenized hyperstroke can capture detailed stroke appearances, including shape and color variations. Based on the quality of the composition of all 328 strokes (Fig. 3 (b)), we conclude that hyperstroke can successfully encode the opacity of strokes from timelapse contexts, enabling the reproduction of artistic illustrations with a much more condensed representation.

Assistive Sketch Generation. We explore the transformer model proposed in Section 2.2 to predict subsequent stroke sequences from user-provided contexts. Given the challenges of scaling up transformers to learn practical artistic drawing sequences due to the scarcity of large-scale incremental drawing datasets, we instead conduct a proof-of-concept using the *Quick, Draw!* dataset [Ha and Eck 2017] and show results in Figure 4. Given canvas context and text conditions, the model demonstrates the ability to generate visually pleasing, temporally intuitive, and coherent sketching sequences that compose meaningful doodles. This generation can be performed unconditionally or prompted from additional user-provided hyperstrokes.

4 Conclusion

In this work, we propose hyperstroke, an efficient and expressive stroke representation designed to capture the essence of artistic drawing strokes. It is particularly well-suited for transformer-based sequential modeling. In the future, we may aim to investigate better hyperstroke encoding schemes, the balance between canvas encodings and historic stroke inputs, and conduct more comprehensive assistive drawing evaluations, by which we believe that the representational capabilities of hyperstroke will inspire future HCI applications in assistive drawing. It will enable a more comprehensive understanding of artistic drawing techniques and fulfill the genuine needs of artists, thereby enhancing their productivity.

Acknowledgments

This work is funded by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. UGC/FDS11/E02/23). We would also like to thank Hao Chen and Linda Wei for providing the invaluable demo and training data.

References

Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M Hospedales, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. 2020. Pixelor: A Competitive Sketching AI Agent. So you think you can sketch? *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 12873–12883.

David Ha and Douglas Eck. 2017. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477* (2017).

Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. 2021. Paint transformer: Feed forward neural painting with stroke prediction. In *Proceedings of the IEEE/CVF international conference on computer vision.* 6598–6607.

Yotam Nitzan, Zongze Wu, Richard Zhang, Eli Shechtman, Daniel Cohen-Or, Taesung Park, and Michaël Gharbi. 2024. Lazy Diffusion Transformer for Interactive Image Editing. *arXiv preprint arXiv:2404.12382* (2024).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning.* PMLR, 8748–8763.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 10684–10695.

Jaskirat Singh, Cameron Smith, Jose Echevarria, and Liang Zheng. 2021. Intelli-paint: Towards developing human-like painting agents. *arXiv preprint arXiv:2112.08930* (2021).

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 586–595.

Ningyuan Zheng, Yifan Jiang, and Dingjiang Huang. 2018. Strokenet: A neural painting environment. In *International Conference on Learning Representations.*