

Interactive Neural Painting

Elia Peruzzo ^{a,*}, Willi Menapace ^a, Vidit Goel ^b, Federica Arrigoni ^c, Hao Tang ^d, Xingqian Xu ^e, Arman Chopikyan ^b, Nikita Orlov ^b, Yuxiao Hu ^b, Humphrey Shi ^{b,e}, Nicu Sebe ^a, Elisa Ricci ^{a,f}

^a University of Trento, Italy

^b Picsart AI Research (PAIR), United States of America

^c Politecnico di Milano, Italy

^d ETH Zürich, Switzerland

^e University of Illinois Urbana Champaign (UIUC), United States of America

^f Fondazione Bruno Kessler, Italy

ARTICLE INFO

Communicated by Nikos Paragios

MSC:
41A05
41A10
65D05
65D17

Keywords:
Neural painting
Generative models
Interactive generation

ABSTRACT

In the last few years, Neural Painting (NP) techniques became capable of producing extremely realistic artworks. This paper advances the state of the art in this emerging research domain by proposing the first approach for Interactive NP. Considering a setting where a user looks at a scene and tries to reproduce it on a painting, our objective is to develop a computational framework to assist the user's creativity by suggesting the next strokes to paint, that can be possibly used to complete the artwork. To accomplish such a task, we propose I-Paint, a novel method based on a conditional transformer Variational AutoEncoder (VAE) architecture with a two-stage decoder. To evaluate the proposed approach and stimulate research in this area, we also introduce two novel datasets. Our experiments show that our approach provides good stroke suggestions and compares favorably to the state of the art.

1. Introduction

One of the main objectives of image generation methods is enabling novel and more powerful ways in which humans can express their creativity. This objective inspired a lot of research and advancements in deep generative models, that are now able to produce outputs with photorealistic quality in several generation tasks. A recent trend in deep image generation is that of improving the way in which users can control and interact with the generation process, thus providing tools to convey users intentions. In this context, recent works allow users to generate or edit images with high quality by sketching (Ghosh et al., 2019; Liu et al., 2021a), modifying the semantic layouts (Lee et al., 2020; Ling et al., 2021; Park et al., 2019; Zhu et al., 2020), or providing a text prompt (Bau et al., 2021; Nichol et al., 2021; Ramesh et al., 2021; Xu et al., 2022). These methods, however, allow users to influence the final output only in an indirect manner, *i.e.* through the sketched semantic layout or the input text.

Recently, several learning-based methods for painting generation have been proposed, commonly referred to as Neural Painting (NP) methods. Differently from other generative approaches that operates in the pixel space, NP methods leverage a parameterized brushstroke representation which is more aligned to how humans visualize and

conceptualize an artwork (Kotovenko et al., 2021; Liu et al., 2021b; Zou et al., 2021). The strokes-based vector representation offers several benefits compared to the pixel-based representation, such as the ability to modify or erase individual strokes. Additionally, separating the representation from the rendering process enables the strokes to be rendered at any desired output resolution.

Painting has historically been a powerful tool with which humans expressed their creativity. However, in this respect, current NP methods are inherently limited, as they are only designed to reconstruct and stylize a given target image, leaving no possibility for the user to influence the generation process. Lacking the ability to integrate users' painting style, these methods are unsuitable in interactive scenario. This work represents the first attempt to fill this gap in the literature and bring the next level of interaction to NP. Inspired by *en plein air* painting, *i.e.* the setting where a painter looks at an outdoor scene and tries to represent it on a canvas, we introduce the novel task of Interactive Neural Painting (see Fig. 1). Specifically, we propose an iterative and interactive process where, given a reference image the user would like to paint and an incomplete canvas, a computational tool based on NP techniques assists the user in drawing the painting. The tool provides multiple suggestions about the next strokes at each iteration, from which the user can choose to continue its artwork. Such

* Corresponding author.

E-mail address: elia.peruzzo@unitn.it (E. Peruzzo).

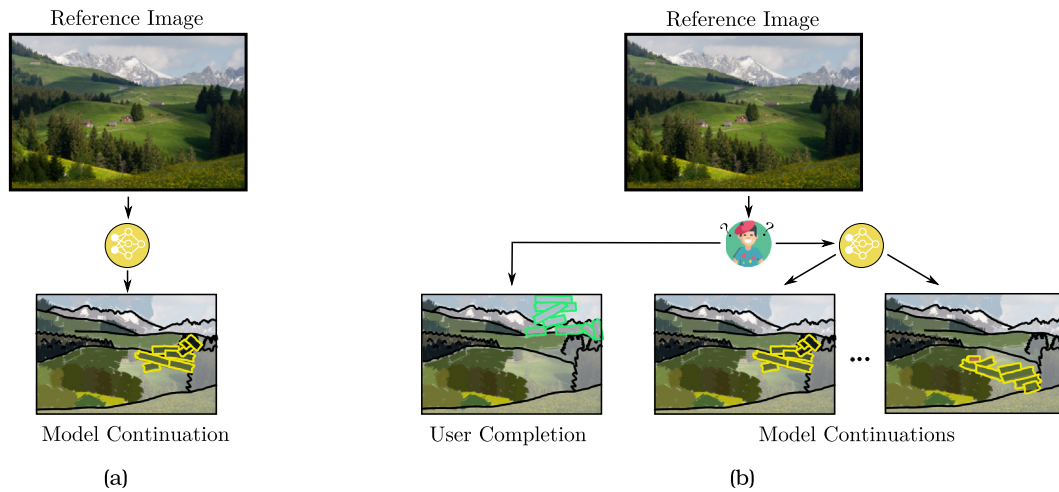


Fig. 1. Inspired by en-plain-air setting, we tackle the task of representing a reference image on the canvas. The final painted result is obtained by sequentially placing strokes on the canvas, refining the artwork until the user is satisfied. In this figure we describe a single step of such a process. In Neural Painting (a), a deep architecture is learned in order to create a realistic artwork of a reference image with a sequence of strokes (outlined in yellow). Note that **the user is not involved** in this process. Differently, in Interactive Neural Painting (b), **the user directly contributes** to the generation process in an interactive loop with the deep model. At each iteration, the model provides a set of stroke suggestions based on the reference image and current state of the painting, and the user either selects a suggested set of strokes (outlined in yellow) or directly draws new strokes on the canvas (outlined in green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

a tool speeds up the painting process but, differently from existing NP approaches, leaves the user a high degree of control on the final output, with the potential of making painting an artistic medium accessible not only to highly-skilled individuals. Our system can be integrated into digital drawing tools used by amateur and professional artists, such as Adobe Photoshop, GIMP, and Krita, as shown by our demo in the *Supp. Mat.*

We devise the first method for Interactive NP (INP). Our method, which we call I-Paint, introduces a conditional transformer VAE architecture that generates stroke suggestions. To ensure seamless interaction with the user, our method is specifically trained to produce stroke suggestions that closely match the dynamics of the painting process represented in a given dataset of painting demonstrations. The dataset is built to reflect in a synthetic manner the main aspects valued by a human painter such as color consistency, local proximity and object-based painting. Additionally, artists typically begin by portraying a rough depiction of the reference image, and incrementally incorporate finer details during the painting process (Zhao et al., 2020; Singh et al., 2021). We follow previous work (Zou et al., 2021; Liu et al., 2021b) and adopt a coarse-to-fine assumption in our dataset to reflect this behavior, with rough strokes spatially covering the canvas in the first stages of the painting and detailed localized strokes towards the final stages (see Section 4.1 for more details). To effectively learn the characteristics of the stroke dataset, we introduce a distribution matching loss that minimizes the discrepancies between the suggested strokes and the painting demonstrations. In addition, a two-stage VAE decoder is proposed that tightly integrates visual features into the stroke prediction process. Furthermore, we make our approach probabilistic by nature to capture the complex distribution of possible continuations given the current canvas state. In this way, I-Paint can produce multiple suggestions about what to paint next. We demonstrate our method on two novel datasets which we specifically introduce for the INP task, built upon the *ADE 20K Outdoor* (Zhou et al., 2017) and *Oxford-IIIT Pet* (Parkhi et al., 2012) datasets. Our extensive evaluation shows that our model produces a wide set of suggestions that closely match the characteristics of the painting demonstrations. Quantitative comparison against state-of-the-art NP methods, supported by results on a user study, demonstrates state-of-the-art performance of our method.

Contributions. To summarize, our main contributions are:

- The novel image generation task of INP, which for the first time brings interactivity to neural painting.

- The first approach based on conditional transformer VAE to address this task, with specific architectural choices and training protocols.
- Two novel synthetic datasets and a set of evaluation metrics for training and evaluating INP models, to foster and assess the research in this new area.

2. Related work

In this section, we discuss the most related works in the field of NP and interactive image generation.

2.1. Neural painting

Neural Painting techniques are derived from the intriguing idea of teaching machines how to paint. NP is typically formalized as the process of artistically recreating a given image using a neural network which generates a series of strokes. Several approaches are present in the literature that address this task. Some of them make use of reinforcement learning (RL) (Huang et al., 2019; Schaldenbrand and Oh, 2021; Singh et al., 2021; Singh and Zheng, 2021; Xie et al., 2012), where, given the current environment represented by the present status of the canvas and a reference image, an agent is trained to predict the parameters of the next strokes. The training objective is formulated as the maximization of the cumulative rewards of the whole painting process, typically expressed as the increase in similarity between the new canvas state and the reference image. Since no gradients need to be directly backpropagated from the reward function, RL-based methods do not require a differentiable stroke rendering procedure.

Other methods, instead, make use of a differentiable stroke renderer that allows direct optimization of a loss objective. Among these methods, Zou et al. (2021) and Kotovenko et al. (2021) directly optimize a set of parameters describing the stroke sequence, producing high-quality results at the cost of long inference times. Other works overcome this limitation by using a model to predict stroke parameters rather than directly optimizing them. In this context, the state of the art is represented by Paint Transformer (PT, Liu et al., 2021b), where NP is expressed as a set prediction problem and a transformer-based architecture is proposed that predicts the parameters of a stroke set with a feedforward network. Our method shares similarities with Liu et al. (2021b) as we also assume a differentiable stroke renderer and predict stroke parameters with a feedforward network. However, we

address a different (and new) task, by focusing on an *interactive* setting that requires seamless integration between model predictions and user inputs.

Finally, a different class of methods focuses on the closely related task of sketch generation which consists in the generation of abstract sketches. Sketch-RNN (Ha and Eck, 2018) and Sketch-BERT (Lin et al., 2020) represent sketches as sequences of points and are based respectively on RNN and transformer models. While these methods can be employed in interactive tasks such as sketch completion, they are not able to reproduce natural images and do not model realistic painting effects.

2.2. Interactive image generation

Interactive image generation refers to the task of automatically generating photo-realistic images, conditioned on user inputs. Early works fall into two directions: (1) image-to-image translation, which investigates the problem of translating an input image to a target domain, allowing to synthesize photos from label maps or reconstruct objects from edge maps (Isola et al., 2017; Tang et al., 2019; Zhu et al., 2017a,b); (2) learning a human-interpretable latent space (Chen et al., 2016), projecting a natural image into it, manipulating the latent code to achieve an edit, and synthesizing a new image accordingly (Abdal et al., 2019, 2020; Brock et al., 2017; Lin et al., 2021; Zhu et al., 2016). To provide a more compelling experience, recent works on interactive image generation allow more user-friendly interaction, e.g. by means of sketches (Ghosh et al., 2019; Liu et al., 2021a), semantic maps (Lee et al., 2020; Ling et al., 2021; Park et al., 2019; Tang et al., 2020; Zhu et al., 2020), paint strokes (Cheng et al., 2022; Singh et al., 2022), and text (Bau et al., 2021; Nichol et al., 2021; Ramesh et al., 2021; Xu et al., 2022). Ghosh et al. (2019) introduced iSketchNFill, an interactive GAN-based sketch-to-image translation method that helps novice users to easily create images of simple objects with a sparse sketch and the desired object category. Differently, GauGAN (Park et al., 2019) converts a semantic segmentation mask to a photo-realistic image with a spatially-adaptive normalization layer. To flexibly manipulate an existing image, Bau et al. (2019) allows the user to perform a localized edit of an image by selecting a specific region, while (Liu et al., 2021a) empowers the user to edit low-level details by sketching the desired modifications. Singh et al. (2022) condition the image generation on strokes painted by the user, to provide a more intuitive way compared to segmentation maps, while Cheng et al. (2022) rely on sketches and paint strokes to guide the generation process, allowing both flexibility and precise control. Fueled by the success of text-to-image generation (Ramesh et al., 2021), very recent works proposed to control image manipulations with natural language, creating an intuitive way of interaction for the user. Notably, Jiang et al. (2021) and Shi et al. (2021) focused on the problem of global image editing, while Xia et al. (2021) proposed a unique framework to both generate and manipulate images using text inputs.

However, all the aforementioned methods are evaluated by the quality of the generated results, the diversity of the suggestions, and how closely they match the users' input. The proposed task of INP adds an additional level of complexity. Since INP gives the users complete stroke-by-stroke control over the final artwork, it is necessary to represent the *process* that leads to the final result. To ensure smooth interaction with the user, the method should follow a paint-like-demonstration behavior (see Section 3 for a discussion). This requirement, and the level of control over the final output, is peculiar to the task of INP, and differentiates it from the existing literature in interactive image generation.

3. Methodology

In the following we describe our method in detail: Section 3.1 provides a formalization of the task and an overview of the proposed method, Section 3.2 describes the architectural components, respectively the context encoder, the VAE encoder and decoder, Section 3.3 illustrates the employed losses and training procedure, Section 3.4 describes the inference process, while Section 3.5 describes the implementation details.

3.1. Problem formulation and overview

We start the section by formalizing the task of Interactive Neural Painting (INP). We assume a dataset D of reference images paired with a sequence of stroke parameters $s = s^{1:T}$ of length T , representing a decomposition of the image into a sequence of individual strokes. Each stroke is represented by a tuple of eight parameters as $s = (x, \rho, \sigma, \omega)$, where x is the position of the stroke center on the canvas, ρ is the color, σ represents the stroke size expressed as height and width, and ω is the stroke orientation. At time t , given a reference image I_{ref} and the corresponding sequence of strokes up to the current time $s^{1:t}$, the INP task consists in predicting a set of stroke sequences of length k . The set of predicted stroke sequences is presented to the user who can either select one sequence, partially or in its entirety, as the painting continuation or manually define the next strokes if no proposed sequence captures the user's current painting intentions (see Fig. 1). Note that predicting a sequence of length $k > 1$ (Liu et al., 2021b; Zou et al., 2021) gives the user the possibility to better understand whether the proposed continuation corresponds to her painting intentions. The operation is repeated iteratively until completion of the painting. The expected behavior of predicted strokes should exhibit the following characteristics:

- Each sequence makes the canvas more similar to the reference image, hence assisting the user in the final goal of completing the painting.
- Each sequence presents the same characteristics of the dataset stroke sequences in terms of positioning, color, size and orientation, thus ensuring seamless interaction between the user and the painting agent.
- The predicted set contains diverse sequences that cover the main possible continuations of the painting process, hence providing diverse continuation to the user among which to choose.

We devise a set of quantitative evaluation metrics that captures each of these desired behaviors and present them in Section 4.2.

In this paper we propose I-Paint, a method for INP. Our approach consists in a transformer-based conditional VAE architecture and is depicted in Fig. 2. At time t , given a reference image I_{ref} , context strokes $s_c = s^{t-k+1:t}$, and a context image I_c defined as the rendering of strokes $s^{1:t}$, the context encoder C extracts a context vector $c = C(I_{\text{ref}}, I_c, s_c)$. During training, the VAE encoder E encodes the target stroke sequence $s_t = s^{t+1:t+k}$ into a posterior gaussian distribution $\mu_z, \sigma_z^2 = E(s_t, c)$ and the latent code z is sampled from it. During inference, instead, the latent code is sampled from the prior distribution $\mathcal{N}(0, 1)$. The latent code z is used in conjunction with c to condition the VAE decoder that produces the sequence of inferred target strokes $\hat{s}_t = D(z, c)$.

3.2. Architecture components

Next, we describe the architectural components of I-Paint depicted in Fig. 2.

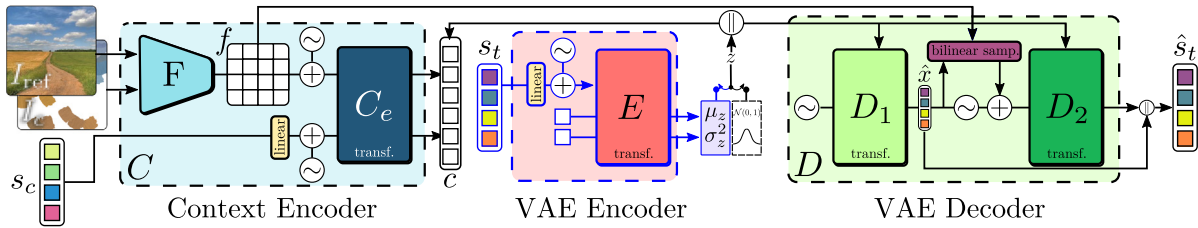


Fig. 2. Overview of I-Paint. Our model is based on a conditional VAE architecture. A context encoder C extracts a context vector c from the reference and context images (I_{ref}, I_c respectively) and the context strokes s_c . The VAE encoder E encodes the target stroke sequence s_t into a posterior distribution. A latent code z is sampled and used in conjunction with c to condition the VAE decoder that reconstructs the target stroke sequence. At inference time, the latent code z is sampled from the prior distribution $\mathcal{N}(0, 1)$. The decoder D is composed of two steps, D_1 which predicts the position of the strokes \hat{x}_t , and D_2 , which predicts the remaining parameters that are concatenated with \hat{x}_t to form the final prediction \hat{s}_t . Transformers (Vaswani et al., 2017) are used to model C , E and D . We use \sim to represent sinusoidal positional encodings and \parallel to represent concatenation. The blue outline denotes components that are used only during training. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Context encoder. The context encoder C receives as input the reference image $I_{ref} \in \mathbb{R}^{3 \times H \times W}$, the context image $I_c \in \mathbb{R}^{3 \times H \times W}$, the sequence of context strokes parameters $s_c \in \mathbb{R}^{k \times 8}$ and produces a representation of the context c . First, a visual feature encoder F , modeled as a CNN, extracts visual features $f_{c, \text{visual}} \in \mathbb{R}^{\text{dim} \times H' \times W'}$ from the input images. Following Liu et al. (2021b), we model F as a separate backbone for reference and context images respectively, and concatenate the output features along the channel dimension to obtain $f_{c, \text{visual}}$. A linear layer is used to extract context stroke features from s_c , resulting in features $f_{\text{strokes}} \in \mathbb{R}^{k \times d_{\text{emb}}}$. Successively, we flatten the spatial dimensions of the visual features $f_{c, \text{visual}} \in \mathbb{R}^{(H' \cdot W') \times d_{\text{emb}}}$, and concatenate them with the strokes features f_{strokes} along the sequence length dimension. The resulting token sequence $c_{in} \in \mathbb{R}^{L \times d_{\text{emb}}}$, with $L = (H' \cdot W') + k$, is enriched with 3D sinusoidal positional encodings (Vaswani et al., 2017) which are added to each element of the sequence. Two encoding dimensions represent the x and y coordinates of the visual feature or stroke and the third is used to represent the temporal position of each stroke in the sequence. Lastly, a Transformer encoder C_e process c_{in} producing the context output $c \in \mathbb{R}^{L \times d_{\text{emb}}}$.

VAE encoder and decoder. We use a VAE conditioned on the context information c to produce a reconstruction \hat{s}_t of the target strokes. We model the VAE encoder E as a transformer decoder receiving as query input a the target strokes $s_t \in \mathbb{R}^{k \times 8}$, which are projected to the hidden dimension of the transformer d_{emb} with a linear layer. The input sequence is enriched with 3D sinusoidal positional encodings (Vaswani et al., 2017), and two learnable tokens corresponding to the output mean and variance of the posterior gaussian distribution $\mu_z, \sigma_z^2 = E(s_t, c)$. We make use of the learnable tokens as a way to obtain outputs from the transformer representing the input sequence pooled over the temporal dimension (Petrovich et al., 2021). The context information is used to condition E and is provided as key and value inputs, conditioning the encoder through cross-attention.

The VAE decoder D produces the sequence of reconstructed target strokes $\hat{s}_t = D(z, c)$. Preliminary experiments show that directly predicting \hat{s}_t from z and the context c is difficult, so we propose a decoder composed of two stages $D = D_2 \circ D_1$, each modeled as a separate transformer decoder. The initial decoder D_1 receives as query input 1D sinusoidal positional encodings (Vaswani et al., 2017) providing temporal information regarding target strokes and predicts stroke positions $\hat{x}_t = D_1(z, c)$ of the target strokes. The transformer decoder is conditioned with the cross-attention mechanism on the context c and on the latent variable $z \sim \mathcal{N}(\mu_z, \sigma_z^2)$ which are received as key and value inputs. The second transformer decoder D_2 is responsible for inferring the remaining stroke parameters conditioned by \hat{x}_t . In order to provide precise information about the reference image in the neighborhood of each predicted position, we extract image features $f_{\hat{x}}$ from f corresponding to the predicted position of each stroke using bilinear sampling, i.e. $f_{\hat{x}} = \text{bilinear}(f, \hat{x}_t)$. Similarly to the VAE encoder, the sampled features are enriched with 3D sinusoidal positional encodings and are used as query inputs to infer the remaining stroke parameters

$(\hat{\rho}_t, \hat{\sigma}_t, \hat{\omega}_t) = D_2(z, c, f_{\hat{x}})$. As with D_1 , c and z condition the decoder as key and value inputs in the cross-attention operation. Finally, the outputs of D_1 and D_2 are combined to form the reconstructed target strokes $\hat{s}_t = (\hat{x}_t, \hat{\rho}_t, \hat{\sigma}_t, \hat{\omega}_t)$.

3.3. Training

We train our model using the β -VAE (Higgins et al., 2017) objective with an isotropic Gaussian prior as the main driving loss:

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{z \sim E(s_t, c)} \|s_t - \hat{s}_t\|_2^2 + \lambda_{\text{KL}} D_{\text{KL}}(\mathcal{N}(\mu_z, \sigma_z^2) \parallel \mathcal{N}(0, 1)). \quad (1)$$

In addition, we notice that imprecisions in the reconstruction of the stroke positions \hat{x}_t may bring to a situation where the reconstructed stroke color $\hat{\rho}_t$ differs from the color of I_{ref} at \hat{x}_t which we call $\tilde{\rho}_t = I_{ref}(\hat{x}_t)$. This mismatch is caused by the model ignoring the reference image and predicting target colors by attending only to context strokes and latent code and leads to performance degradation. For this reason, we introduce a color reconstruction loss that fosters the model to produce output strokes whose color is coherent with I_{ref} :

$$\mathcal{L}_{\text{col}} = \|\tilde{\rho} - \hat{\rho}\|_2^2. \quad (2)$$

Moreover, we propose two additional regularization losses that are aimed at improving the visual results at inference time when the latent codes z are sampled from the prior distribution rather than the posterior. First, we impose the same color reconstruction loss on the predicted strokes to improve color coherency:

$$\mathcal{L}_{\text{col}}^{\text{reg}} = \mathbb{E}_{z \sim \mathcal{N}(0, 1)} \|I_{ref}(x(D(z, c))) - \rho(D(z, c))\|_2^2, \quad (3)$$

where $x(\cdot)$ and $\rho(\cdot)$ represent function extracting respectively position and color from the tuple of stroke parameters. Second, we impose a distribution matching objective aimed at maximizing the similarity between the characteristics of predicted and dataset stroke sequences. In particular, we propose to explicitly maximize the likelihood of sampling from the prior a sequence of strokes that is compatible with the dataset stroke distribution. For each sequence of corresponding context and target strokes, we concatenate them forming vector $s = s_c \parallel s_t = s^{t-k+1:t+k}$, and build the corresponding feature vector ψ capturing the relations between neighboring strokes. The feature vector ψ is computed by taking the concatenation of the stroke features computed as follows:

$$\psi = \parallel_{i=1}^{l_{\text{max}}} (\parallel_{j=1}^{L-l} (s^{i+j} - s^i)), \quad (4)$$

where $L = 2k$ represents the sequence length and l_{max} represents the maximum distance between strokes for which to extract features. In the following, we denote as ψ the features produced on dataset stroke sequences and as $\hat{\psi}$ the features produced on inferred stroke sequences. To make the computation tractable and easy to optimize, we assume independence of each dimension and fit two multivariate

gaussian distributions $\mathcal{N}(\mu_\psi, \Sigma_\psi)$ and $\mathcal{N}(\mu_{\hat{\psi}}, \Sigma_{\hat{\psi}})$, respectively on ψ and $\hat{\psi}$. Successively, we minimize KL divergence between the dataset distribution and the generated strokes distribution:

$$\mathcal{L}_{\text{dist}}^{\text{reg}} = \mathbb{E}_{z \sim \mathcal{N}(0,1)} \mathcal{D}_{\text{KL}}(\mathcal{N}(\mu_{\hat{\psi}}, \Sigma_{\hat{\psi}}) \parallel \mathcal{N}(\mu_\psi, \Sigma_\psi)). \quad (5)$$

The final optimization objective is given by:

$$\mathcal{L} = \mathcal{L}_{\beta\text{-VAE}} + \lambda_{\text{col}} \mathcal{L}_{\text{col}} + \lambda_{\text{col}}^{\text{reg}} \mathcal{L}_{\text{col}}^{\text{reg}} + \lambda_{\text{dist}}^{\text{reg}} \mathcal{L}_{\text{dist}}^{\text{reg}}, \quad (6)$$

where λ_{col} , $\lambda_{\text{col}}^{\text{reg}}$ and $\lambda_{\text{dist}}^{\text{reg}}$ represent positive weighting terms. We show details of our training procedure in the *Supp. Mat.*

3.4. Inference

At inference time, our model is used iteratively to assist users in the creation of paintings corresponding to a reference image I_{ref} . We consider the current state of the canvas as I_c and the last k strokes drawn on the canvas as the context strokes s_c . The context encoder C is used to extract the context representation c . We note that, given the interactive scenario, the context strokes can originate either from the user or by a previous iteration of the model. We then sample a latent vector z from the prior distribution $\mathcal{N}(0, 1)$ and use the decoder to produce a plausible continuation of the painting process $D(z, c)$. By repeatedly sampling the latent vector z from the prior distribution and keeping the context fixed, we provide a diverse set of plausible continuations of the painting from which the user can select the best option or keep drawing strokes manually if not satisfied by the proposals. The process is iterated until the painting is complete.

3.5. Implementation details

Following Liu et al. (2021b), we set the sequence length $k = 8$ in all our experiments. We observe that, at inference time, the effective value of k can be smaller, with the user selecting a sub-sequence of the proposed continuation. Likewise, the length of the context strokes s_c is set to $k = 8$, which can be modified at inference time according to the user needs. We set the image resolution of I_{ref}, I_c to $H = W = 256$, and implement the context feature extractor F as a convolutional network reducing the spatial dimension of the input by a factor of 8, thus resulting in a feature map $f_{c,\text{visual}}$ of size $H' = W' = 16$. This makes the effective length of c_{in} , the input of the context encoder, $L = 256 + 8$. We model the context encoder C_e as a transformer encoder, while the VAE encoder E and VAE decoder D are implemented as transformer decoders. In all the cases, we set the hidden dimension of the models to $d_{\text{emb}} = 256$. We train the final model for 5000 epochs, with a batch size of 32, using the AdamW optimizer (Loshchilov and Hutter, 2019) with initial learning rate of $1e-4$ and cosine scheduler. We select the weights of each loss component in Eq. (6) with a grid-search on the *Oxford-IIIIT Pet INP*, and apply the same configuration for experiments on the *ADE 20K Outdoor INP*. The weight of each loss component is, respectively, $\lambda_{\text{KL}} = 2.5e-4$, $\lambda_{\text{col}} = 2.5e-2$, $\lambda_{\text{col}}^{\text{reg}} = 2.5e-3$ and $\lambda_{\text{dist}}^{\text{reg}} = 5.0e-6$. Additional implementation details are present in the *Supp. Mat.*

4. Experiments

In this section, we perform experimental evaluation of the proposed method for INP. Section 4.1 describes the adopted datasets, Section 4.2 describes the adopted metrics, Section 4.3 shows ablation results on our method, Section 4.4 performs quantitative comparison against baselines and shows qualitative results.

4.1. Datasets

To train our architecture, we assume a dataset of images with an associated sequence of stroke parameters, representing the painting process used to produce the corresponding painting. To produce realistic stroke suggestions, our model captures the characteristics of the painting process represented in the dataset. To overcome the cost associated to collecting human painting demonstrations, we follow recent work of Cheng et al. (2022) and Singh et al. (2022) and choose to demonstrate that our framework is capable of modeling a painting process considering a synthetic dataset of stroke sequences that mimic a human painting process. Importantly, our method is general and learn the characteristics of the strokes provided as demonstration, thus can be readily applied to a human collected dataset if available.

We consider two existing image datasets and associate a sequence of strokes to each image, producing our INP datasets:

- *ADE 20K Outdoor INP*: we employ a subset of 5000 images of the ADE 20K dataset (Zhou et al., 2017) consisting in the set of original images depicting outdoor scenes. We split the dataset into a set of 4750 training images and 250 images for evaluation.
- *Oxford-IIIIT Pet INP*: the dataset consists of 7349 images depicting different cat and dog breeds from *Oxford-IIIIT Pet* (Parkhi et al., 2012), both in indoor and outdoor scenarios. The dataset is split into 6980 training images and 369 images for evaluation.

Each image is decomposed into a sequence of strokes, parameterized as Section 3, using the NP method of Zou et al. (2021). Similar to the human painting process, the obtained sequence is organized in different levels of details, with large strokes depicting the outline of the image first and fine-grained detail later in the sequence. However, such sequences of strokes do not contain the sequential patterns typically produced by humans. Painters, in fact, due to the constraints imposed by physical brushes which discourage changes in color and brush, tend to produce sequences of strokes where the same color and brush sizes are maintained across several subsequent strokes. In addition, it is common for humans to produce paintings on a object-by-object basis (Singh et al., 2021; Zhao et al., 2020) and to produce strokes in contiguous regions. To replicate these patterns in our synthetic dataset and produce sequences with characteristics closer to real ones, we perform a reordering of the stroke sequence produced by Zou et al. (2021) by optimizing a cost function that penalizes sequences with large differences in size, position or color between adjacent strokes or where adjacent strokes are placed on different subjects. Specifically, we perform a reordering of the sequences by minimizing the following cost function, computed along the complete stroke sequence:

$$\text{cost} = \sum_{t=2}^T (\lambda_x^{\text{ord}} \|x^t - x^{t-1}\|_2^2 + \lambda_\rho^{\text{ord}} \|\rho^t - \rho^{t-1}\|_2^2 + \lambda_\sigma^{\text{ord}} (\sigma^t - \sigma^{t-1})^2 + \lambda_{\text{obj}} \chi(x^t, x^{t-1})) \quad (7)$$

where λ_x^{ord} , $\lambda_\rho^{\text{ord}}$, $\lambda_\sigma^{\text{ord}}$, λ_{obj} are positive weighting parameters. The function $\chi(x^t, x^{t-1})$ is equal to 1 if the input strokes are located on different subjects and 0 otherwise, and it is computed using the dataset segmentation masks. We ensure that the ordering relation between overlapping strokes is preserved, guaranteeing that both the original and reordered sequences of strokes produce the same visual output when rendered. Such problem is an instance of the Sequential Ordering Problem which we optimize following Helsgaun (2017).

4.2. Evaluation metrics

As outlined in Section 3.1, the key three factors that we expect in the INP setting are: (i) the method is painting the reference image, (ii) the produced strokes parameters have characteristics that match the ones of the stroke dataset, (iii) diverse stroke continuations can be produced for the same context. We devise a set quantitative metrics to capture these desiderata, and describe them in the following:

Table 1
Loss ablation results on the ADE 20K Outdoor INP dataset and the Oxford-IIIT Pet INP dataset.

	$\mathcal{L}_{\beta\text{-VAE}}$	\mathcal{L}_{col}	$\mathcal{L}_{\text{col}}^{\text{reg}}$	$\mathcal{L}_{\text{dist}}^{\text{reg}}$	ADE 20K Outdoor INP					Oxford-IIIT Pet INP						
					L2↓	FSD↓	FVD↓	WD↓	DTW↓	LPIPS↑	L2↓	FSD↓	FVD↓	WD↓	DTW↓	LPIPS↑
✓	✗	✗	✗	✗	0.136	1.64	11.9	0.032	0.849	0.038	0.155	1.29	13.2	0.031	0.851	0.038
✓	✓	✗	✗	✗	0.058	2.44	7.18	0.034	0.899	0.031	0.057	2.05	7.31	0.033	0.910	0.029
✓	✓	✓	✗	✗	0.043	6.84	8.16	0.040	0.974	0.028	0.039	5.17	6.77	0.035	0.942	0.030
✓	✓	✗	✓	✓	0.094	1.92	10.5	0.033	0.892	0.044	0.091	1.16	9.51	0.031	0.893	0.039
Full	✓	✓	✓	✓	0.044	2.04	6.60	0.034	0.893	0.033	0.042	1.51	6.72	0.032	0.893	0.030

- Stroke Color L2 (L2) (i): we measure the L2 difference between the color of the predicted strokes and the color of the underlying reference image region. To avoid big strokes from dominating the metric, the L2 distance associated to each stroke is normalized by the stroke area before averaging.
- Fréchet Stroke Distance (FSD) (ii): inspired by FID (Heusel et al., 2017), we introduce a metric measuring the similarity between ground truth and predicted stroke sequences. For each sequence of context and target strokes, we compute stroke features ψ as in Eq. (4). We report the Fréchet Distance (Fréchet, 1957) between the distribution of features derived from ground truth sequences and predicted ones.
- Fréchet Video Distance (FVD) (Unterthiner et al., 2018) (ii): given a sequence of context and target strokes, we generate videos of the corresponding canvas rendered up to each stroke in the sequence. We use FVD between videos produced with ground truth and predicted strokes as a metric capturing the similarity between sequences.
- Wasserstein Distance (WD) (Kantorovich, 1939) (ii): following Liu et al. (2021b), we adopt the Wasserstein Distance between gaussian distributions fitted on the ground truth and predicted strokes as a stroke reconstruction quality metric.
- Dynamic Time Warping (DTW) (Mueller, 2007) (ii): we employ DTW between the ground truth and inferred target strokes to measure the quality of matching.
- LPIPS (Zhang et al., 2018) (iii): following Zhu et al. (2017b), we use LPIPS as a metric to compute the diversity of the produced outputs. For each reference image and context, we produce 5 stroke predictions and measure the average LPIPS diversity between all pairs of rendered results.

For each image in the test set, we extract 5 sequences of corresponding context and target strokes and compute the metrics on these samples. We note that WD and DTW require paired sequences of ground truth and reconstructed target sequences, while at inference time our method generates plausible stroke sequences that may not match the ground truth. For these metrics, we adopt a top-1 sampling strategy (Yu et al., 2021) and generate 100 plausible stroke sequences, reporting the metric obtained for the best one.

4.3. Ablation study

In this section, we ablate the main losses and architectural components of the proposed method. To improve the number of analyzed model configurations, in this section we reduce the number of training epochs to 1000. We start our analysis by ablating the contribution of the proposed losses (see Table 1). Training the model only with the β -VAE loss produces stroke outputs with high diversity but whose Stroke Color L2 is the highest in all the configurations, suggesting that the model is predicting strokes that are not consistent with the reference image. Introducing \mathcal{L}_{col} promotes the model to take I_{ref} into account, resulting in a consistent reduction of the Stroke Color L2. To further improve the performance, we introduce our two training regularization losses, which are aimed at improving quality when the latent code z is sampled from the prior distribution at inference time. Introducing $\mathcal{L}_{\text{col}}^{\text{reg}}$ improves color accuracy as demonstrated by

the best Stroke Color L2, but prevents the model from learning the users’ painting style resulting in the highest FSD. Vice-versa, with only our proposed $\mathcal{L}_{\text{dist}}^{\text{reg}}$ we can effectively learn the distribution of strokes, achieving the best performances in terms of FSD, WD, and LPIPS, but performance decreases in terms of the Stroke Color L2. Only when combining all the proposed losses in our full model we obtain good performance under *all* metrics.

Next, to ablate the contribution of each proposed architectural component, we produce the following modified versions of our method: (i) remove the context information provided by s_c and I_c , the only context information comes from the reference image I_{ref} ; (ii) remove C_e ; (iii) remove the two-step decoding procedure and replace it with a single transformer decoder that directly predicts \hat{s}_i ; We show the ablation results in Table 2. As expected, removing s_c and I_c from the context information significantly degrades the performance. Interestingly, LPIPS is the highest among the configurations, probably because, without conditioning from the context, the predictions can vary more freely. Likewise, removing the transformer encoder block C_e consistently reduces the metrics, showing the importance of the module that provides richer context information to the decoder D by combining visual and strokes features. Configuration (iii) shows the impact of the two-step decoding procedure, designed to provide the model with richer visual information. This version of the method results in a degraded Stroke Color L2, FSD and FVD, showing the importance of detailed visual features in the prediction of strokes.

4.4. Comparison against baselines

In this section, we compare our method against the state-of-the-art NP methods of Zou et al. (2021) and of Liu et al. (2021b). Due to the novelty of the task, these works are not directly comparable with the proposed one since they do not consider interaction, and need to be adapted to the INP setting. A key component of the selected methods that makes them unsuitable for INP is their hierarchical rendering pipeline that iteratively divides the reference image and the canvas into smaller regions and operates on each in separation. This procedure allows the models to progressively focus on finer details and accurately reconstruct the reference image, but produces poor performance in INP since at each iteration the method may be forced to output strokes in a region far from the area the user is painting. To avoid this limitation and make the model aware of the context, instead of operating on hardcoded regions, at each iteration we consider as the current region a portion of the image centered on the last context stroke. We call such region the context region. In this way, we encourage the method to output a sequence of strokes in the neighborhood of the context. In addition, we make the size of the region proportional to the area of the context strokes s_c , fostering the models to produce strokes with a level of detail compatible with the one currently adopted by the user. We apply this modification to both methods and produce the following baselines:

- Paint Transformer (PT) (Liu et al., 2021b): the model makes a prediction in the context region, but no explicit knowledge about the distribution of the dataset stroke sequences can be leveraged.

Table 2

Ablation architectural choices on *ADE 20K Outdoor INP* and *Oxford-IIIT Pet INP* datasets. Each row represent a different model version, respectively: (i) remove context information I_C and s_c , (ii) remove C_c , (iii) decode all strokes parameters with a single-step decoder and our full model.

Model version	<i>ADE 20K Outdoor INP</i>						<i>Oxford-IIIT Pet INP</i>					
	L2↓	FSD↓	FVD↓	WD↓	DTW↓	LPIPS↑	L2↓	FSD↓	FVD↓	WD↓	DTW↓	LPIPS↑
(i)	0.060	16.4	21.7	0.058	1.11	0.039	0.059	11.9	23.3	0.058	1.13	0.037
(ii)	0.050	2.27	7.45	0.035	0.907	0.031	0.048	1.87	7.78	0.036	0.915	0.027
(iii)	0.052	2.28	7.12	0.032	0.868	0.029	0.049	1.70	7.83	0.033	0.893	0.028
Full	0.044	2.04	6.60	0.034	0.893	0.033	0.042	1.51	6.72	0.032	0.893	0.030

Table 3

Comparison with baselines on *ADE 20K Outdoor INP* and *Oxford-IIIT Pet INP* datasets.

Method	<i>ADE 20K Outdoor INP</i>						<i>Oxford-IIIT Pet INP</i>					
	L2↓	FSD↓	FVD↓	WD↓	DTW↓	LPIPS↑	L2↓	FSD↓	FVD↓	WD↓	DTW↓	LPIPS↑
PT (Liu et al., 2021b)	0.056	10.6	9.06	0.073	1.41	0	0.048	11.3	8.77	0.074	1.47	0
SNP (Zou et al., 2021)	0.044	13.7	7.05	0.082	1.25	0.018	0.037	16.3	6.09	0.075	1.27	0.017
SNP+ (Zou et al., 2021)	0.045	8.50	7.20	0.081	1.16	0.017	0.039	9.57	5.95	0.074	1.20	0.017
I-Paint	0.040	1.50	6.27	0.031	0.876	0.032	0.037	1.12	4.73	0.029	0.867	0.031

- Stylized Neural Painting (SNP) (Zou et al., 2021): we randomly initialize the sequence of predicted strokes to lie in the context region and optimize their parameters with the original SNP objective. Note that the method is not expressly conditioned on the context strokes and does not consider the characteristics of the dataset stroke sequences.
- Stylized Neural Painting+ (SNP+) (Zou et al., 2021): we modify SNP to explicitly take into consideration the characteristics of the dataset stroke sequence. In detail, we modify the SNP optimization objective by introducing a term similar to the distribution matching loss $\mathcal{L}_{\text{dist}}^{\text{reg}}$ to produce stroke sequences whose features $\hat{\psi}$ match those of the training dataset ψ . We extract stroke features from the dataset using Eq. (4) and fit a multivariate gaussian distribution with independent components on ψ . We improve the realism of inferred strokes by maximizing the likelihood that the inferred features $\hat{\psi}$ match the fitted distribution.

Table 3 shows comparison results of our method against the baselines. To ensure a fair comparison, we follow Liu et al. (2021b) and set the length of predicted sequence to $k = 8$ for all the methods. While baseline methods are designed with the main objective of producing strokes that closely match the reference image, we notice that our method presents a Stroke Color L2 metric similar to SNP and SNP+, and lower with respect to PT. Moreover, Paint Transformer and Stylized Neural Painting have, by design, no way to leverage information about the characteristics of the dataset strokes and tend to produce stroke sequences whose characteristics do not match the ones in the dataset. This is highly reflected in the metrics that capture the ability to paint like the demonstration; our method strongly outperforms PT and SNP in terms of the WD, DTW, FVD, and FSD. On the other side, SNP+ can exploit such information. As expected, this greatly reduces the FSD compared to the naive SNP, but comes at the cost of increasing the Stroke Color L2. Interestingly, our method can outperform SNP+ in this metric, suggesting that better performance of our model is due not only to distribution matching objectives but also to the architecture design. Finally, we evaluate the capacity of the method to produce varied plausible outputs for a fixed context. We observe that, while PT is deterministic and no variability can be produced, diverse predictions for a given context can be obtained for SNP and SNP+ by starting the optimization from different randomly initialized stroke parameters. Our method instead is probabilistic by nature, with a conditional VAE designed to generate different plausible continuations for a fixed context. Despite the non-determinism of some baselines, the diversity of their predictions is inferior to the ones obtained with our method, which achieves the highest LPIPS diversity score.

Table 4

User study comparing the preferences between I-Paint and the respective baseline.

	PT Liu et al. (2021b)	SNP Zou et al. (2021)	SNP+ Zou et al. (2021)
Preferences	97.9%	97.1%	95.0%

User study. We complement our quantitative results with a user study. We show the users a set of reference videos with rendered stroke sequences from the dataset followed by two videos, one with rendered strokes produced by one of the baselines, and one produced by I-Paint. To ease the evaluation, we produce sequences with a length of 24 strokes and render the obtained strokes in a short video. We ask the users to express which of the two videos has strokes whose characteristics resemble the reference videos the most. We gather a total of 960 votes from 8 unique users. We report the results in Table 4, showing a clear preference for I-Paint when evaluated on the INP task (see *Supp. Mat.* for additional details).

4.5. Qualitative results

We provide qualitative results comparing our method with the baselines. Visualizing results as still images provides limited understanding of the INP task, we refer the reader to *Supp. Mat.* for video results along with a working demo. When comparing different methods, the predicted strokes sequences should satisfy three criteria: (i) make the canvas more similar to the reference image, (ii) present similar characteristics as the demonstrations, (iii) provide diverse continuations Sections (Section 3.1, 4.2). While all the methods perform similarly when assessing (i), I-Paint strongly outperform the competitors according to (ii) and (iii) which are peculiar of the *interactive* component of the task.

First, we analyze the ability of I-Paint to model the characteristics of the dataset strokes (ii). Given the reference image and the context fixed, we sample 100 possible stroke continuations. A method that correctly captures the original stroke distribution is expected to yield at least a stroke continuation close to the ground truth. In Fig. 3 we plot qualitative results showing the sampled sequence that better matches the ground truth in terms of L2 distance. In all examples, our method is able to produce a sequence close to the ground truth, while PT, SNP and SNP+ struggle to generate a matching sequence, indicating that our method is able to better capture the characteristics of the dataset stroke sequences. In Fig. 4 we show the heatmap depicting the probability, obtained with the aforementioned sampling procedure, that a certain

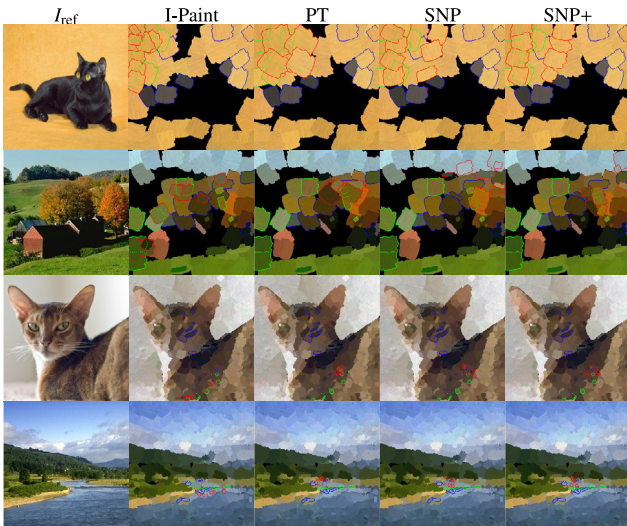


Fig. 3. Qualitative comparison of I-Paint with baselines. Given the **context** (blue) strokes, we generate 100 **predicted** (red) stroke sequences and plot the one that better matches the **ground truth** (green). First row, only I-Paint is able to produce a sequence whose stroke positioning is similar to the ones of the ground truth, while the baselines tend to unrealistically cluster the strokes in a tight area. Second row, successive strokes predicted by I-Paint have similar colors as in the dataset demonstrations, while the baselines unrealistically jump between the grass, the sky, and the trees. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

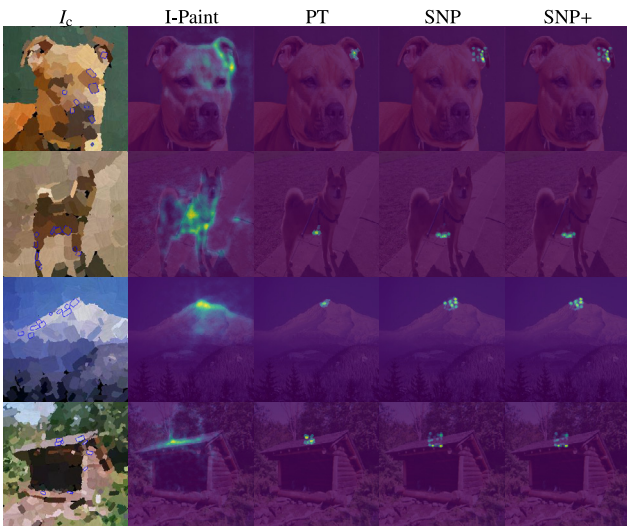


Fig. 4. Next predicted stroke probability distribution. For each method, we show the probability that a given pixel will be occupied by a predicted stroke for the given context. The context (outlined in blue in the first column) is kept fixed and we sample $n = 500$ continuations for each method to estimate the probability distribution. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

pixel will be covered by one of the next k strokes. We notice that only I-Paint is able to produce different plausible suggestions (iii) while predicting strokes on the same object as the last context strokes (ii). Second, we further evaluate the ability of I-Paint to predict different stroke continuations given a fixed context (iii). Note that, as shown in Fig. 1(b), this is an important feature of an INP method since at each iteration of the painting process the method should be able to propose at least a stroke sequence that matches the user painting intentions. In Fig. 5 we show different stroke suggestions for a fixed context obtained by sampling different latent codes z from the unit normal prior distribution $\mathcal{N}(0, 1)$. Our method is capable of generating diverse stroke

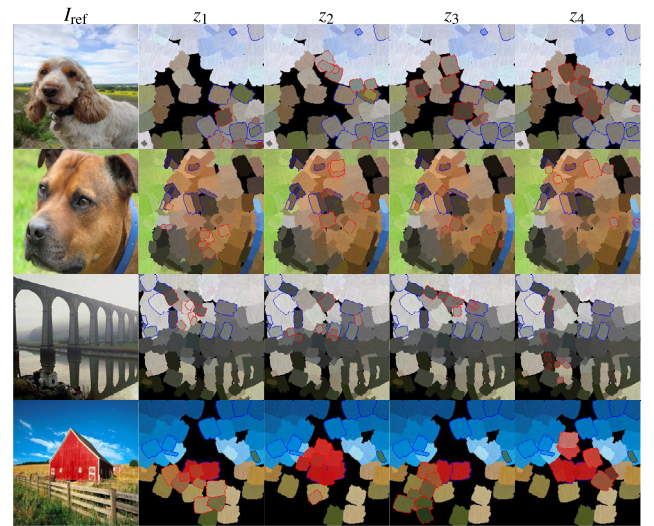


Fig. 5. Diversity of proposed continuations. Given a fixed **context** (blue), we sample different $z_i \sim \mathcal{N}(0, 1)$ and plot the correspondent **predicted** (red) strokes in different columns. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

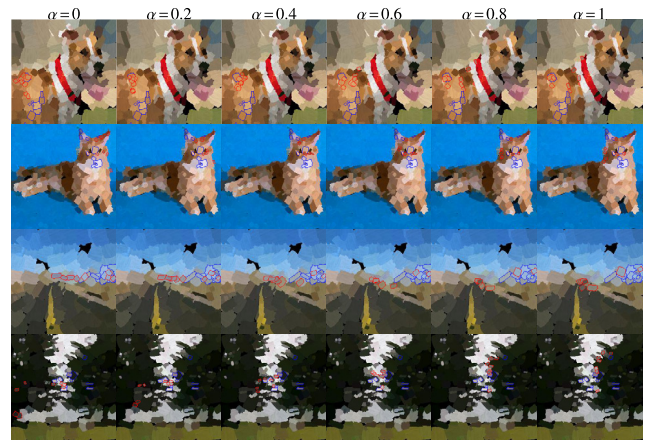


Fig. 6. Latent code z interpolation results (the reference image is omitted for better visualization). For each image, given the same **context** strokes, we sample two latent codes $z_{\text{start}}, z_{\text{end}} \sim \mathcal{N}(0, 1)$ and linearly interpolate, obtaining $z_i = (1 - \alpha) \cdot z_{\text{start}} + \alpha \cdot z_{\text{end}}$ (outlined in red). The strokes smoothly transition changing position but focusing on the same object. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

continuations, each of which focuses on a similar region, color and subject with respect to the given context. Finally, we show the structure of the learned latent representation (Fig. 6). Specifically, we sample two different latent codes from the prior distribution, $z_{\text{start}}, z_{\text{end}} \sim \mathcal{N}(0, 1)$ and we linearly interpolate between the two samples, plotting the predicted results along the interpolation path (see *Supp. Mat.* for video animation). It is possible to notice that the strokes smoothly transition between the two samples $z_{\text{start}}, z_{\text{end}}$, changing their position but focusing on the same subject suggesting that the learned latent space is well structured.

5. Conclusions

In this paper we introduce the novel task of interactive neural painting, where a user tries to reproduce a scene on a painting and the objective is to give (multiple) suggestions about the next strokes to paint, thus helping the user in producing its artwork. The proposed method is based on a conditional transformer VAE architecture, which

is demonstrated on two novel datasets. Our experiments show that our model: is correctly painting the reference image, outputs strokes whose characteristics are close to demonstration data, gives diverse yet plausible brush-stroke suggestions; and outperforms the analyzed baselines on a large set of metrics and on a user study. We hope that our work can stimulate further research in this domain.

CRedit authorship contribution statement

Elia Peruzzo: Conceptualization, Methodology, Software, Investigation, Writing. **Willi Menapace:** Conceptualization, Methodology, Writing. **Vidit Goel:** Software, Investigation. **Federica Arrigoni:** Conceptualization, Methodology, Writing. **Hao Tang:** Conceptualization, Data curation. **Xingqian Xu:** Conceptualization. **Arman Chopikyan:** Conceptualization. **Nikita Orlov:** Conceptualization. **Yuxiao Hu:** Conceptualization, Supervision. **Humphrey Shi:** Supervision, Resources. **Nicu Sebe:** Supervision, Resources, Funding acquisition, Writing. **Elisa Ricci:** Supervision, Project administration, Writing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

The research was supported by the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU and the PRIN project CREATIVE (Prot. 2020ZSL9F9).

Appendix

In this appendix, we provide additional details about parametrization and rendering of the strokes, a description of the architecture and the losses, and user study details.

This appendix is complemented by a supplementary website where we show additional qualitative results in the form of videos. In particular, we showcase a demo of I-Paint in an Interactive Neural Painting (INP) scenario, where a user interacts with our model to paint a reference image. The material is available at this link: <https://helia95.github.io/inp-website>.

Dataset

Due to the cost associated with the acquisition of a real stroke dataset, we evaluate our model on two synthetic stroke datasets, built to mimic human painting style (see Section 4.1 of the main paper for a discussion). Here we detail the dataset acquisition procedure. Examples from the dataset are provided on the supplementary website.

Images. We rely on two publicly available datasets, each containing images and associated segmentation masks. The *Oxford-IIIT Pet* dataset (Parkhi et al., 2012), contains 7349 images of cats and dogs of different species for a total of 37 different classes. The *ADE 20K* dataset (Zhou et al., 2017) is a large-scale dataset containing 20,100 images from more than 3K classes. The dataset is filtered to contain images of outdoor scenes, resulting in a subset of 5000 images.

Strokes Parametrization. Following the work of Zou et al. (2021), we parametrize the strokes as $s = (x, \rho, \sigma, \omega)$. The center of the stroke is represented by $x = (x_x, x_y)$. The height and width of the strokes are represented by $\sigma = (\sigma_h, \sigma_w)$, while ω represents the orientation, which is the counter-clock wise angle in the range $[0, \pi]$. Lastly, the color

of the stroke is represented by $\rho = (\rho_r, \rho_g, \rho_b)$. All the parameters are normalized to lie in the interval $[0, 1]$.

Decomposition. We make use of Stylized Neural Painting (SNP) (Zou et al., 2021), to extract a sequence of brushstrokes from a given image. We notice that SNP tends to produce very large strokes in the first iterations of the method, which cover a wide area of the canvas. This practice is unrealistic since the size limitations of physical brushstrokes would prevent a human painter from doing this. To circumvent such behavior, we clamp the parameter σ to a maximum value of 0.4. As described in Zou et al. (2021), we employ a progressive rendering pipeline with a total of 4 iterations, dividing the image in a grid with 4, 9, 16, and 25 regions. We allocate a different number of strokes during the progressive rendering process, respectively 30, 20, 15, and 10 to each region, which results in a total of 790 strokes per image. Lastly, SNP represents the color of each stroke using two triples of (r, g, b) values that are interpolated to obtain a smooth color. For simplicity, we use the average of the two, and represent the stroke with a uniform color ρ .

Reordering. We perform a reordering of the sequences, by minimizing the cost function described in Section 4.1 of the main paper.

Rendering. To render the strokes on the canvas, we follow Liu et al. (2021b) and use a *parameter free* renderer. Starting from a primitive brushstroke, affine transformations are applied to obtain the foreground I_{s^t} and the alpha matte α_{s^t} associated to $s^t = (x, \rho, \sigma, \omega)$. The canvas can be updated computing $I_c^t = \alpha_{s^t} \cdot I_{s^t} + (1 - \alpha_{s^t}) \cdot I_c^{t-1}$. We refer the reader to Liu et al. (2021b) for additional details.

Method

Architecture. We report more details of the architecture, depicted in Fig. 2 of the main paper. Our model relies on the Transformer architecture of Vaswani et al. (2017), where we set the embedding dimensionality $d_{\text{emb}} = 256$, the number of heads in multi-head attention to 4, the dimension of the intermediate linear layer to 1024, and the dropout rate to 0. The CNN encoder F is composed of 4 convolutional blocks with residual connection and receives as input an image of size 256×256 . The spatial resolution of the features is reduced by a factor of 2 in each block, resulting in a 16×16 output feature map. Following Liu et al. (2021b), we use two distinct image encoders for I_{ref} and I_c . The features obtained by the two input images are concatenated and projected to the embedding dimensionality d_{emb} . Similarly, the context strokes s_c and the target strokes s_t are projected to d_{emb} using a linear layer. The remaining components are implemented as standard transformers blocks. In particular, C_e is transformer encoder with number of layers equal to 8, while E , D_1 , D_2 are transformer decoders with number of layers equal to 6.

Losses. We provide additional details about the computation of the losses. The reconstruction loss component of $\mathcal{L}_{\beta\text{-VAE}}$ is computed by weighting the reconstruction error differently for each component of the stroke parameter:

$$\|s_t - \hat{s}_t\|_2^2 = \lambda_x \|x_t - \hat{x}_t\|_2^2 + \lambda_\rho \|\rho_t - \hat{\rho}_t\|_2^2 + \lambda_\sigma \|\sigma_t - \hat{\sigma}_t\|_2^2 + \lambda_\omega \|\omega_t - \hat{\omega}_t\|_2^2 \quad (\text{A.1})$$

with $\lambda_x = 1$, $\lambda_\rho = 2.5e-1$, $\lambda_\sigma = 1$ and $\lambda_\omega = 1$. In early experiments, we noticed that the component corresponding to the predicted color ρ , i.e. $\|\rho_t - \hat{\rho}_t\|_2^2$, was difficult to jointly optimize with \mathcal{L}_{col} , hence we reduced its weights until convergences of the two. The last component of our objective is the distribution matching loss $\mathcal{L}_{\text{dist}}^{\text{reg}}$. We noticed that, when this loss is used, the predicted strokes may present a distorted height/width ratio. To avoid this issue, when computing this loss we exclude the size σ and the orientation ω from the computation of features ψ . The same modification is applied to the SNP+ baseline for fairness of comparison.

Experiments

User study. We now provide details on the user study presented in the main paper. Each task of the user study consists of an HTML page divided into two sections. In the first, called the demonstration section, we show a collection of stroke sequences taken from the training set. In the second section, called the evaluation section, we show the users two stroke sequences produced from the same reference images and stroke context, one produced with I-Paint and the other with one of the baselines. We asked the participants to select which of the two sequences of strokes presents characteristics (in terms of stroke positions, colors, and subject consistency) that are most similar to the ones of the strokes in the demonstration section. To ease the evaluation, we produce sequences with a length of 24 strokes and render the obtained strokes in a short video. The user study was conducted on 40 images taken from the test set of *Oxford-IIIT Pet INP* dataset, from which a total of 120 tasks was generated. We collected a total of 960 votes from 8 unique users. Examples of the user study are provided on the supplementary website.

References

- Abdal, R., Qin, Y., Wonka, P., 2019. Image2stylegan: How to embed images into the stylegan latent space? In: CVPR.
- Abdal, R., Qin, Y., Wonka, P., 2020. Image2stylegan++: How to edit the embedded images? In: CVPR.
- Bau, D., Andonian, A., Cui, A., Park, Y., Jahanian, A., Oliva, A., Torralba, A., 2021. Paint by word. arXiv preprint arXiv:2103.10951.
- Bau, D., Strobel, H., Peebles, W., Wulff, J., Zhou, B., Zhu, J., Torralba, A., 2019. Semantic photo manipulation with a generative image prior. ACM Trans. Graph. (Proc. ACM SIGGRAPH).
- Brock, A., Lim, T., Ritchie, J.M., Weston, N., 2017. Neural photo editing with introspective adversarial networks. In: ICLR.
- Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., Abbeel, P., 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In: NeurIPS.
- Cheng, S.-I., Chen, Y.-J., Chiu, W.-C., Lee, H.-Y., Tseng, H.-Y., 2022. Adaptively-realistic image generation from stroke and sketch with diffusion model.
- Fréchet, M., 1957. Sur la distance de deux lois de probabilité. C. R. Hebd. Seances Acad. Sci. 244 (6), 689–692.
- Ghosh, A., Zhang, R., Dokania, P.K., Wang, O., Efros, A.A., Torr, P.H., Shechtman, E., 2019. Interactive sketch & fill: Multiclass sketch-to-image translation. In: ICCV.
- Ha, D., Eck, D., 2018. A neural representation of sketch drawings. In: 6th International Conference on Learning Representations, ICLR.
- Helsgaun, K., 2017. An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problems. Roskilde University, pp. 24–50.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2017. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: NeurIPS.
- Higgins, I., Matthey, L., Pal, A., Burgess, C.P., Glorot, X., Botvinick, M.M., Mohamed, S., Lerchner, A., 2017. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In: ICLR.
- Huang, Z., Heng, W., Zhou, S., 2019. Learning to paint with model-based deep reinforcement learning. In: ICCV.
- Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: CVPR.
- Jiang, W., Xu, N., Wang, J., Gao, C., Shi, J., Lin, Z., Liu, S., 2021. Language-guided global image editing via cross-modal cyclic mechanism. In: CVPR.
- Kantorovich, L.V., 1939. The mathematical method of production planning and organization. In: Management Science. pp. 363–422.
- Kotovenko, D., Wright, M., Heimbrecht, A., Ommer, B., 2021. Rethinking style transfer: From pixels to parameterized brushstrokes. In: CVPR.
- Lee, C.-H., Liu, Z., Wu, L., Luo, P., 2020. Maskgan: Towards diverse and interactive facial image manipulation. In: CVPR.
- Lin, H., Fu, Y., Jiang, Y.-G., Xue, X., 2020. Sketch-BERT: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt.
- Lin, J., Zhang, R., Ganz, F., Han, S., Zhu, J.-Y., 2021. Anycost gans for interactive image synthesis and editing. In: CVPR.
- Ling, H., Kreis, K., Li, D., Kim, S.W., Torralba, A., Fidler, S., 2021. EditGAN: High-precision semantic image editing. In: NeurIPS.
- Liu, S., Lin, T., He, D., Li, F., Deng, R., Li, X., Ding, E., Wang, H., 2021b. Paint Transformer: Feed forward neural painting with stroke prediction. In: ICCV.
- Liu, H., Wan, Z., Huang, W., Song, Y., Han, X., Liao, J., Jiang, B., Liu, W., 2021a. Deflconet: Deep image editing via flexible low-level controls. In: CVPR.
- Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization. In: ICLR.
- Mueller, M., 2007. Dynamic time warping. In: Information Retrieval for Music and Motion. Springer, pp. 69–84.
- Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M., 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741.
- Park, T., Liu, M.-Y., Wang, T.-C., Zhu, J.-Y., 2019. GauGAN: Semantic image synthesis with spatially adaptive normalization. In: SIGGRAPH.
- Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V., 2012. Cats and dogs. In: CVPR.
- Petrovich, M., Black, M.J., Varol, G., 2021. Action-conditioned 3D human motion synthesis with transformer VAE. In: ICCV.
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I., 2021. Zero-shot text-to-image generation. In: ICML.
- Schaldenbrand, P., Oh, J., 2021. Content masked loss: Human-like brush stroke planning in a reinforcement learning painting agent. In: AAAI.
- Shi, J., Xu, N., Xu, Y., Bui, T., Démoncourt, F., Xu, C., 2021. Learning by planning: Language-guided global image editing. In: CVPR.
- Singh, J., Smith, C., Echevarria, J.I., Zheng, L., 2021. Intelli-paint: Towards developing human-like painting agents. ArXiv.
- Singh, J., Zheng, L., 2021. Combining semantic guidance and deep reinforcement learning for generating human level paintings. In: CVPR.
- Singh, J., Zheng, L., Smith, C., Echevarria, J., 2022. Paint2Pix: Interactive painting based progressive image synthesis and editing. arXiv preprint arXiv:2208.08092.
- Tang, H., Xu, D., Sebe, N., Wang, Y., Corso, J.J., Yan, Y., 2019. Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In: CVPR.
- Tang, H., Xu, D., Yan, Y., Torr, P.H., Sebe, N., 2020. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In: CVPR.
- Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S., 2018. Towards accurate generative models of video: A new metric & challenges. arXiv:1812.01717.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: NeurIPS.
- Xia, W., Yang, Y., Xue, J.-H., Wu, B., 2021. Tedigan: Text-guided diverse face image generation and manipulation. In: CVPR.
- Xie, N., Hachiya, H., Sugiyama, M., 2012. Artist agent: A reinforcement learning approach to automatic stroke generation in oriental ink painting. IEICE Trans. Inf. Syst. 96-D, 1134–1144.
- Xu, Z., Lin, T., Tang, H., Li, F., He, D., Sebe, N., Timofte, R., Van Gool, L., Ding, E., 2022. Predict, prevent, and evaluate: Disentangled text-driven image manipulation empowered by pre-trained vision-language model. In: CVPR.
- Yu, Y., Zhan, F., Wu, R., Pan, J., Cui, K., Lu, S., Ma, F., Xie, X., Miao, C., 2021. Diverse image inpainting with bidirectional and autoregressive transformers. In: Proceedings of the 29th ACM International Conference on Multimedia. Association for Computing Machinery, New York, NY, USA, pp. 69–78.
- Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR.
- Zhao, A., Balakrishnan, G., Lewis, K.M., Durand, F., Guttag, J.V., Dalca, A.V., 2020. Painting many pasts: Synthesizing time lapse videos of paintings. In: CVPR.
- Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A., 2017. Scene parsing through ADE20K dataset. In: CVPR.
- Zhu, P., Abdal, R., Qin, Y., Wonka, P., 2020. Sean: Image synthesis with semantic region-adaptive normalization. In: CVPR.
- Zhu, J.-Y., Krähenbühl, P., Shechtman, E., Efros, A.A., 2016. Generative visual manipulation on the natural image manifold. In: ECCV.
- Zhu, J.-Y., Park, T., Isola, P., Efros, A.A., 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: CVPR.
- Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E., 2017b. Toward multimodal image-to-image translation. In: NeurIPS.
- Zou, Z., Shi, T., Qiu, S., Yuan, Y., Shi, Z., 2021. Stylized neural painting. In: CVPR.