Creative Painting with Latent Diffusion Models

Xianchao Wu
NVIDIA
xianchaow@nvidia.com, wuxianchao@gmail.com

Abstract

Artistic painting has achieved significant progress during recent years. Using a variational autoencoder to connect the original images with compressed latent spaces and a cross attention enhanced U-Net as the backbone of diffusion, latent diffusion models (LDMs) have achieved stable and high fertility image generation. In this paper, we focus on enhancing the creative painting ability of current LDMs in two directions, textual condition extension and model retraining with Wikiart dataset. Through textual condition extension, users' input prompts are expanded with rich contextual knowledge for deeper understanding and explaining the prompts. Wikiart dataset contains 80K famous artworks drawn during recent 400 years by more than 1,000 famous artists in rich styles and genres. Through the retraining, we are able to ask these artists to draw artistic and creative paintings on modern topics. Direct comparisons with the original model show that the creativity and artistry are enriched.

1 Introduction

Artistic painting has achieved significant progress during recent years thanks to the appearing of hundreds of GAN variants (Jabbar et al., 2020; Wang et al., 2021). However, adversarial training has been reported to be notoriously unstable and can lead to mode collapse. To escape from adversarial training and inspired by non-equilibrium thermodynamics, diffusion probabilistic models (Sohl-Dickstein et al., 2015), such as noise-conditional score network (NCSN) (Song and Ermon, 2019), denoising diffusion probabilistic models (DDPM) (Ho et al., 2020), stable diffusion models in latent spaces (Rombach et al., 2021) have achieved GAN-level sample quality without adversarial training. These diffusion models are appealing with rather flexible model architectures, exact log-likelihood computation, and inverse problem solving without re-training models.

There are two Markov chain style processes in a typical diffusion model. The first process is a *forward diffusion process* which appends multiple-scale random noise to a given data sample "step by step" or "in jump" until the disturbed sample slip into a predefined isotropic Gaussian distribution. This process does not include trainable parameters. The second process is a *reverse diffusion process* which generates a target distribution data sample from pure noise guided by some (user-input) pre-given conditions. A parameterized deep learning model is required in this reverse process.

Intuitively speaking, the forward diffusion process can be recognized as "directional blasting of a building" $\mathbf{x}_0$ to "ruins with dusts" $\mathbf{x}_T$. The learning algorithm is a *reverse engineering* which learns how to (re-)construct a building (expressed by $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ with a parameter set θ and $t \in \{1, ..., T\}$) from each step of *inverse* directional blasting (expressed by $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$) of each given building sample $\mathbf{x}_0$. In one step of this reverse engineering, $\mathbf{x}_{t-1}$ represents "one complete wall" in a building and $\mathbf{x}_t$ represents "concrete and sands" that can be used to construct the complete wall $\mathbf{x}_{t-1}$ in a reconstruction process or can be obtained from the complete wall $\mathbf{x}_{t-1}$ in a forward "blasting" process. The reconstruction process is learned from the blasting process with targets such as noise prediction in DDPM (Ho et al., 2020) or score prediction using score matching strategy in NCSN (Song and Ermon, 2019).

We follow a recent impressive work of high-resolution image synthesis with LDMs by given textual or visual conditions[1] (Rombach et al., 2021). There are several proposals in this LDM. The first proposal is applying the encoder part of a pre-trained variational autoencoder to project images into low-dimension latent spaces and then perform

[1] https://github.com/CompVis/stable-diffusion

Proceedings of the 2nd Workshop on When Creative AI Meets Conversational AI (CAI2), COLING 2022, pages 59–80
October 12–17, 2022.

diffusion/construction processes. Training diffusion models on such a low-dimension representation space allows us to reach a near-optimal point between computation complexity reduction and detail preservation to boost virtual fidelity of constructed images. The second is a cross-attention-enhanced (Vaswani et al., 2017) U-Net framework (Ronneberger et al., 2015) in the diffusion model where general conditioning inputs such as text or bounding boxes are taken as *memory* (i.e., keys and values in the cross-attention layers) for the query (latent representations of images to be generated) to retrieve information on. Finally, the decoder module in the variational autoencoder is applied to recover the target image into high-resolution.

We aim at improving the *creativity* of image synthesis, or painting, using conditional LDMs. It is relatively difficult to precisely define the concept of creativity since it is subjective and influenced by culture, history, and region. The color, style, objects included in painting reflect rich emotions of numerous topics. For example, when we are given a textual condition, "a painting of a virus monster playing guitar", we can recognize noun entities such as "virus monster" and "guitar" and a verbal action "playing". What are the emotions involved in this textual hint? Happy, surprise and funny should be the major emotions. The painting requires less imagination since we should better include the entries with a determined action.

However, there are challenges for the models to draw painting for rather high-level topics such as "urbanization of China" or "Asian morning". These textual hints should be enriched and extended with concrete objects and actions to tell a story in a painting or in a series of paintings. Extensions to "urbanization of China" include "originally a collection of fishing villages, Shenzhen rapidly grew to be one of the largest cities in China", "a train runs on the snow-capped mountains of the Qinghai-Tibet Plateau", and "left-behind children running in wheat-field". Given an initial textual hint, we leverage Wikipedia and large-scale pretrained language models to execute this extension.

In addition, we retrain existing checkpoints by the WikiArt paintings dataset[2] which has a collection of 81,444 fine-art paintings from 1,119 artists, ranging from fifteenth century to modern times. This dataset contains 27 different styles (e.g., *Minimalism*, *Symbolism*, *Realism*) and 45 different genres. As far as our knowledge, it is currently the largest digital art datasets publicly available for research usage. This dataset was used to train an ArtGAN (Tan et al., 2017) where conditions such as categorical label information was used for artwork synthesis. In this paper, we embed the textual information of artists, year, styles, and genres as additional conditions to the LDM. Through this way, we can explicitly invite Vincent van Gogh or Rembrant to help us "draw" artworks of modern topics such as "urbanization of China".

This paper is organized as follows. In Section 2, we briefly review the background knowledge required for understanding the stable diffusion models (Rombach et al., 2021). In particular, we describe the two processes defined in DDPM (Ho et al., 2020), the variational autoencoder framework and loss functions used in it (Esser et al., 2020), cross attention enhanced U-Net which acts as the backbone of the diffusion model, and pseudo numerical methods integrated with DDIMs for fast sampling. In Section 3, we describe our proposal of extending users' prompts by pretrained language models and existing knowledge resources. In Section 4, we show detailed information of the Wikiart dataset and our pipeline of retraining. We describe the experiments in Section 5 and finally conclude in Section 6.

2 Background

Diffusion models have been successfully used in image generation (Rombach et al., 2021), text-to-speech synthesis (Popov et al., 2021; Jeong et al., 2021), sing synthesis and conversion (Liu et al., 2021; Xue et al., 2022), music generation (Mittal et al., 2021) and healthcare Medical Anomaly Detection (Wolleb et al., 2022). Surveys can be find in (Croitoru et al., 2022; Cao et al., 2022; Yang et al., 2022).

We limit our discussion to text-to-image generation by leveraging the LDMs (Rombach et al., 2021) and existing checkpoints[3]. We briefly review the core processes and target objectives of DDPMs (Ho et al., 2020) that are used in LDMs. In addition, variational autoencoders enhanced with KL-divergence, cross-attention embedded U-Net (Ronneberger et al., 2015; Vaswani et al., 2017), CLIP pretrained language models (Radford et al.,

[2] https://www.wikiart.org/ and can be downloaded from https://archive.org/download/wikiart-dataset/wikiart.tar.gz

[3] https://huggingface.co/CompVis/stable-diffusion-v-1-4-original

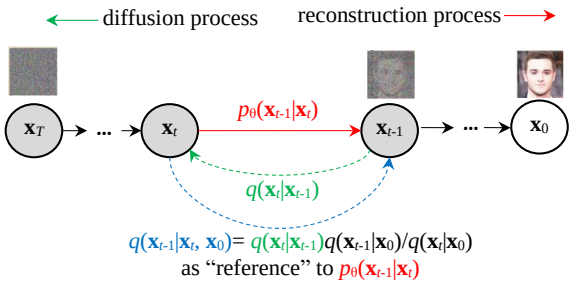

Figure 1: The Markov chain of forward diffusion (backward reconstruction) process of generating a sample by step-by-step adding (removing) noise. Image adapted from (Ho et al., 2020).

2021) and sampling algorithms such as that used in denoising diffusion implicit models (DDIMs) (Song et al., 2020) and pseudo numerical methods (Liu et al., 2022) will be briefly reviewed.

2.1 DDPM

Given a data point $\mathbf{x}_0$ sampled from a real data distribution $q(\mathbf{x})$ ($\mathbf{x}_0 \sim q(\mathbf{x})$), Ho et al. (2020) define a *forward diffusion process* in which small amount of Gaussian noise is added to sample $\mathbf{x}_0$ in T steps to obtain a sequence of noisy samples $\mathbf{x}_0, ..., \mathbf{x}_T$. A predefined (hyper-parameter) variance schedule $\{\beta_t \in (0,1)\}_{t=1}^T$ controls the step sizes:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}); \quad (1)$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}). \quad (2)$$

When $T \to \infty$, $\mathbf{x}_T$ is equivalent to following an isotropic Gaussian distribution. Note that, there are no trainable parameters used in this forward diffusion process.

Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, we can express an arbitrary step t's diffused sample $\mathbf{x}_t$ by the initial data sample $\mathbf{x}_0$:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t. \quad (3)$$

Here, noise $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$ shares the same shape with $\mathbf{x}_0$ and $\mathbf{x}_t$.

In order to reconstruct from a Gaussian noise input $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, we need to learn a model p_θ to approximate the conditional probabilities to run the *reverse diffusion process*:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)); \quad (4)$$

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t). \quad (5)$$

Note that the reverse conditional probability is tractable by first applying Bayes' rule to three Gaussian distributions and then completing the "quadratic component" in the $\exp(\cdot)$ function:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}) \quad (6)$$

$$= q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \quad (7)$$

$$\propto \exp(-\frac{1}{2\tilde{\beta}_t}(\mathbf{x}_{t-1} - \tilde{\boldsymbol{\mu}}_t)^2). \quad (8)$$

Here, variance $\tilde{\beta}_t$ is a scalar and mean $\tilde{\boldsymbol{\mu}}_t$ depends on $\mathbf{x}_t$ and noise $\boldsymbol{\epsilon}_t$:

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t; \quad (9)$$

$$\tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t). \quad (10)$$

Intuitively, $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ acts as a *reference* to learn $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$. We can use the variational lower bound (VLB) to optimize the negative log-likelihood:

$$-\log p_\theta(\mathbf{x}_0) \leq -\log p_\theta(\mathbf{x}_0) + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)). \quad (11)$$

Using the definitions of $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ in Equation 2 and $p_\theta(\mathbf{x}_{0:T})$ in Equation 5, a loss item L_t ($1 \leq t \leq T-1$) is expressed by:

$$L_t = D_{\mathrm{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \quad (12)$$

$$= \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}_t}\left[\frac{1}{2 \parallel \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) \parallel_2^2} \parallel \tilde{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) \parallel^2\right].$$

We further reparameterize the Gaussian noise term instead to predict $\boldsymbol{\epsilon}_t$ from time step t's input $\mathbf{x}_t$ and use a simplified objective that ignores the weighting term:

$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T], \mathbf{x}_0, \boldsymbol{\epsilon}_t}\left[\parallel \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \parallel^2\right] \quad (13)$$

$$= \mathbb{E}\left[\parallel \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t) \parallel^2\right].$$

In (Rombach et al., 2021), LDMs are proposed so that the diffusion processes are performed in compressed latent spaces through a pretrained variational autoencoder $\mathcal{E}(\mathbf{x}_0)$:

$$L_t^{\text{LDM}} = \mathbb{E}_{\mathbf{z}_0=\mathcal{E}(\mathbf{x}_0), \boldsymbol{\epsilon}_t, t}\left[\parallel \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t) \parallel^2\right] \quad (14)$$

$$= \mathbb{E}\left[\parallel \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t) \parallel^2\right].$$

In order to perform condition-based image synthesis, a pre-given textual prompt (or other formats

such as layout) y is first encoded by a domain specific encoder $\tau_\theta(y)$ and then sent to the model to predict $\boldsymbol{\epsilon}_\theta$:

$$L_t^{\text{LDM}} = \mathbb{E}_{\mathcal{E}(\mathbf{x}_0),\boldsymbol{\epsilon}_t,t}\left[\| \boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, t, \tau_\theta(y)) \|^2\right]. \tag{15}$$

Here, $\tau_\theta(y)$ acts as memory (key and value) in the cross-attention mechanism (Vaswani et al., 2017) and can be jointly trained together with $\boldsymbol{\epsilon}_\theta$'s U-Net framework (Ronneberger et al., 2015) from image-conditioning pairs. In the text-to-image generation task of (Rombach et al., 2021), a 12-layer transformer with a hidden dimension of 768 is used[4] (Radford et al., 2021) to encode textual prompts.

2.2 Variational Autoencoder GAN with KL-divergence

The variational autoencoder is pretrained (Esser et al., 2020) beforehand and used directly for encoding the original data sample into latent space and for decoding the reconstructed $\mathbf{z}_0$ back to the original sizes of $\mathbf{x}_0$. In order to combine the effectiveness of the inductive bias of CNNs with the expressivity of transformers, both the encoder ($\mathcal{E}$) and the decoder (or, generator, $\mathcal{G}$) parts of the autoencoder use ResNet blocks and self-attention blocks. Adversarial learning is used to train this vector quantised GAN framework with a combination of several losses:

(1) a *reconstruction* loss:

$$\mathcal{L}_{\text{rec}} = \| \mathbf{x} - \mathcal{G}(\mathbf{q}(\mathcal{E}(\mathbf{x}))) \|^2, \tag{16}$$

where $\mathbf{q}(\cdot)$ is element-wise quantization in (Esser et al., 2020) and a simple 2D 1×1 convolution network in the stable diffusion implementation. We set $\hat{\mathbf{x}} = \mathcal{G}(\mathbf{q}(\mathcal{E}(\mathbf{x}))$ hereafter.

(2) a *perceptual* loss using the learned perceptual image patch similarity (LPIPS) loss (Zhang et al., 2018):

$$\begin{aligned}\text{Scale}(\mathbf{x}) &= (\mathbf{x} - \text{shift})/\text{scale},\\ g_i(\mathbf{x}) &= \| \text{VGG}_i(\text{Scale}(\mathbf{x})) \|_2,\\ \mathcal{L}_{\text{per}} &= \sum_{i=0}^{4} \left\{\text{lin}_i((g_i(\mathbf{x}) - g_i(\hat{\mathbf{x}}))^2)\right\}.\end{aligned} \tag{17}$$

Here, "shift" and "scale" respectively stands for mean vector and standard deviation vector of each channel of the images in the training data. A pretrained VGG checkpoint[5] is used here and VGG_i stands for the i-th layer's output tensor with half-size down sampling shapes (e.g., h, w=256, 128, 64, 32, 16 and c=64, 128, 256, 512, 512). A group of "dropout + conv2d 1×1" (linear) modules lin_i are used project the mean square distances of $\mathbf{x}$ and $\hat{\mathbf{x}}$ into channel number of 1 and then average on height and width. The five scale losses are added up together as the final perceptual loss.

(3) a KL loss between the diagonal Gaussian distribution constructed from $\mathbf{q}(\mathcal{E}(\mathbf{x}))$ = $[\boldsymbol{\mu}; \log\boldsymbol{\sigma}^2]$ and $\mathcal{N}(0, \mathbf{I})$:

$$\begin{aligned}\mathcal{L}_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \| \mathcal{N}(0, \mathbf{I})) =\\ \sum_{c,h,w} (\boldsymbol{\mu}^2 + \boldsymbol{\sigma}^2 - 1 - \log\boldsymbol{\sigma}^2)/2,\end{aligned} \tag{18}$$

where c is channel number, h is height and w is width for an image. The output tensor $\mathbf{q}(\mathcal{E}(\mathbf{x}))$ is separated into two parts (e.g., from (6, 64, 64) to two (3, 64, 64) shape tensors) for the mean and the log of the variance of the Gaussian distribution.

(4) GAN losses which includes the following component:

$$\mathcal{L}_g = -\log\mathcal{D}(\hat{\mathbf{x}}), \tag{19}$$

$$\mathcal{L}_d = \text{Hinge}(\mathcal{D}(\mathbf{x}), \mathcal{D}(\hat{\mathbf{x}})) \tag{20}$$

$$= \frac{\text{relu}(1 - \mathcal{D}(\mathbf{x})) + \text{relu}(1 + \mathcal{D}(\hat{\mathbf{x}}))}{2}. \tag{21}$$

Here, $\mathcal{D}$ stands for a patch-based discriminator that aims to differentiate between real and reconstructed images. Adaptive weight is used to combine these losses and more details can be found in (Esser et al., 2020):

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_1\mathcal{L}_{\text{per}} + \lambda_2\mathcal{L}_{\text{KL}} + \lambda_3\mathcal{L}_g + \lambda_4\mathcal{L}_d. \tag{22}$$

In the configuration used in this paper, $\lambda_1 = 1.0$, $\lambda_2 = 1e-06$. Specially,

$$\lambda_3 = \frac{\nabla_{\mathcal{G}_{\text{last}}}[\mathcal{L}_{\text{rec}} + \lambda_1\mathcal{L}_{\text{per}}]}{\nabla_{\mathcal{G}_{\text{last}}}[\mathcal{L}_g] + \delta}. \tag{23}$$

Here, $\nabla_{\mathcal{G}_{\text{last}}}$ stands for the gradient of the combined reconstruction and perceptual losses with respect to the last layer of $\mathcal{G}$, and $\delta = 1e-4$ is used for numerical stability. The model sets $\lambda_3 = \lambda_4 = 0.0$ at the first M (e.g., 50,000) iterations to focus on training the reconstruction and perceptual abilities of the model. After M iterations, λ_4 is set to be 1.0 for adversarial learning.

[4] https://huggingface.co/openai/clip-vit-large-patch14

[5] https://download.pytorch.org/models/vgg16-397923af.pth

2.3 U-Net with Cross Attention

In (Rombach et al., 2021), a U-Net with a multi-head cross attention mechanism (Vaswani et al., 2017) is used to predict $\boldsymbol{\epsilon}_\theta$ with a MSE loss for training (Equation 15). In a typical U-Net implementation, there are five blocks, a *time embedding block* that embeds an input time step t, *input/middle/output blocks* that perform convolutional and self-attention based representations of $\mathbf{z}_t$ and their cross attentions with conditional memory $\tau_\theta(y)$, and finally a *out block* that projects the result tensor back to the shape of $\mathbf{z}_t$.

The *input block* performs a down sampling with a stack of "resnet + spatial transformer" modules (e.g., 12 modules from (channel, height, width) shape of from (4, 64, 64) to (1280, 8, 8)). Then, the *middle block* with "resnet + transformer + resnet" modules links the *input* and *output blocks* without changing the shape of the tensor. Next, the *output block* performs a up sampling with the same number of modules of the input block (e.g., 12 modules from shape (1280, 8, 8) to (320, 64, 64)). There are residual-style shortcut links here: each module's output is sent respectively from the *input block* to the *output block* with the same level. The final *out block* uses a 2D convolutional layer to project the hidden channel number (e.g., 320) back to the original channel number (e.g., 4).

2.4 DDIMs and Pseudo Numerical Methods

DDIMs (Song et al., 2020) generalizes DDPMs via a class of non-Markovian diffusion processes that lead to the same training objective and give rise to implicit models that generate high quality samples much faster. In the non-Markovian forward process, a real vector $\sigma \in \mathbb{R}^T_{\geq 0}$ is introduced to index a family of *inference* distributions:

$$
\begin{aligned}
q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0) &:= q_\sigma(\mathbf{x}_T|\mathbf{x}_0)\prod_{t=2}^{T} q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0);\\
q_\sigma(\mathbf{x}_T|\mathbf{x}_0) &= \mathcal{N}(\sqrt{\bar{\alpha}_T}\mathbf{x}_0,(1-\bar{\alpha}_T)\mathbf{I});\\
q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) &= \mathcal{N}(\tilde{\boldsymbol{\mu}}(\mathbf{x}_0,\mathbf{x}_t,\sigma_t),\sigma_t^2\mathbf{I});\\
\tilde{\boldsymbol{\mu}}(\mathbf{x}_0,\mathbf{x}_t,\sigma_t) &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \\
&\quad \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}.
\end{aligned}
$$

The mean function $\tilde{\boldsymbol{\mu}}(\mathbf{x}_0,\mathbf{x}_t,\sigma_t)$ is chosen to ensure that $q_\sigma(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0,(1-\bar{\alpha}_t)\mathbf{I})$ without depending on σ anymore.

In the generative process of DDIM, the *denoised observation* $\mathbf{x}_0$ is predicted from pre-given $\mathbf{x}_t$ (reverse usage of Equation 3):

$$
f_\theta(\mathbf{x}_t,t) := (\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t))/\sqrt{\bar{\alpha}_t}.
$$

Then, a sample $\mathbf{x}_{t-1}$ can be generated from $\mathbf{x}_t$ via:

$$
\begin{aligned}
\mathbf{x}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}f_\theta(\mathbf{x}_t,t)\\
&\quad + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,t)+\sigma_t\boldsymbol{\epsilon}_t. \quad (24)
\end{aligned}
$$

When $\sigma_t = 0$ for all t, the coefficient of $\boldsymbol{\epsilon}_t$ becomes zero and samples are generated from $\mathbf{x}_T$ to $\mathbf{x}_0$ with a fixed procedure. The $\text{DDIM}(\cdot)$ is thus defined as:

$$
\mathbf{x}_{t-1}, f_\theta(\mathbf{x}_t,t) = \text{DDIM}(\mathbf{x}_t,\boldsymbol{\epsilon}_t,t). \quad (25)
$$

To accelerate the reconstruction process and keep the sample quality, DDIMs (Equation 25) are included in pseudo numerical methods (Liu et al., 2022) which treat DDPMs as solving differential equations on manifolds. In (Rombach et al., 2021)'s code implementation[6] (Algorithm 1), a linear multi-step algorithm, the Adams-Moulton method[7], is used. This pseudo numerical algorithm includes a gradient part of 2nd order pseudo improved Euler, 2nd/3rd/4th order Adams-Bashforth methods, and a transfer part of DDIM. Here, the discrete indices $t-1, t+1$ stand for next (e.g., from T to $T-1$) and former time steps, respectively.

3 Textual Condition Extension

We perform textual condition extension by leveraging wikipedia as the knowledge base and large-scale pretrained language models as implicit knowledge graphs. The pipeline is depicted in Figure 2. Given a textual prompt, we first match it with the title list in wikipedia. At the same time, the input prompt is sent to (1) a pretrained language model, T5 (Raffel et al., 2019), to continue writing by taking the given prompt as a prefix hint and to (2) a pretrained dialog model, DialoGPT[8] (Zhang et al., 2019) that takes the input prompt as "query" and consequently generate "responses".

Wikipedia's titles and contents are used for matching the input prompt and T5/DialoGPT's outputs. We use BM25 (Robertson, 2009) here to

[6] https://github.com/CompVis/stable-diffusion/blob/main/ldm/models/diffusion/plms.py#L218-L232

[7] https://en.wikipedia.org/wiki/Linear_multistep_method#CITEREFHairerN%C3%B8rsettWanner1993

[8] https://github.com/microsoft/DialoGPT

Algorithm 1: Pseudo linear multi-step (PLMS) algorithm enhanced by DDIM

$\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$;
for $t = T, T-1, ..., 1$ **do**
  $e_t = \epsilon_\theta(\mathbf{x}_t, t)$;
  if *t == T* **then**
    # pseudo improved Euler-2nd;
    $\mathbf{x}_{t-1}, f_\theta(\mathbf{x}_t, t) = \text{DDIM}(\mathbf{x}_t, e_t, t)$;
    $e_{t-1} = \epsilon_\theta(\mathbf{x}_{t-1}, t-1)$;
    $e'_t = (e_t + e_{t-1})/2$;
  else if *t == T-1* **then**
    # PLMS-2nd (Adams-Bashforth) ;
    $e'_t = (3e_t - e_{t+1})/2$;
  else if *t == T-2* **then**
    # PLMS-3rd (Adams-Bashforth) ;
    $e'_t = (23e_t - 16e_{t+1} + 5e_{t+2})/12$;
  else
    # PLMS-4th (Adams-Bashforth) ;
    $e'_t = (55e_t - 59e_{t+1} + 37e_{t+2} - 9e_{t+3})/24$;
  $\mathbf{x}_{t-1}, f_\theta(\mathbf{x}_t, t) = \text{DDIM}(\mathbf{x}_t, e'_t, t)$;
return $\mathbf{x}_0$;

simplify the matching process. From the result document(s), we further compute sentence importance to rank their content fertility and the relationship with the initial prompt. We use the (English) text part of LAION-5B[9] and Wikiart to train a TF-IDF model and then use it to score the prompts in the result prompt list. With a higher score, we subjectively believe that the prompt can possibly yield better images. To score the "relationship" with the initial prompt u, we embed a pair of initial and result prompts by T5 and compute their cosine similarity. Thus, the importance of a result prompt v is computed by:

$$w(v) = \text{TFIDF}(v) + \lambda_1 \text{Cos}(\text{T5}(u), \text{T5}(v)). \quad (26)$$

Here, λ_1 stands for a hyper-parameter to balance the scale of two scores.

In addition, we encourage the result prompts to include spatial and temporal information. We leverage a named entity recognizer[10] and regular expressions to recognize place/region names, addresses, time, and date. The number of spatial and temporal entities discounted by a hyper parameter

[9] https://laion.ai/blog/laion-5b/
[10] https://github.com/kamalkraj/BERT-NER

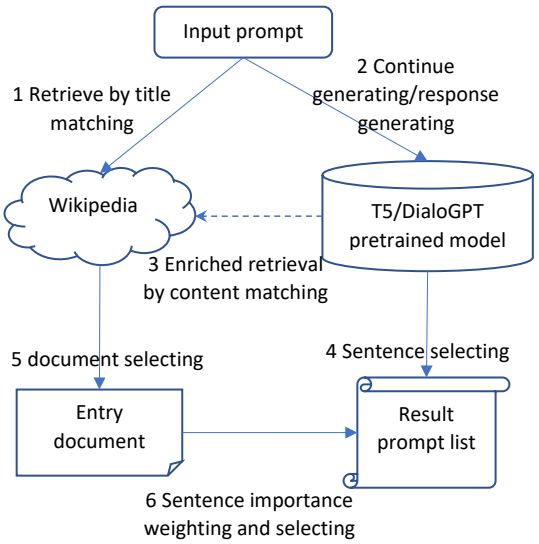

Figure 2: The textual prompt extension pipeline by retrieving wikipedia and continue generating by T5/DialoGPT pretrained language models (Raffel et al., 2019; Zhang et al., 2019).

λ_2 is added with $w(v)$ for the final scoring of a prompt.

4 Retraining with WikiArt

Different artists have quite different numbers of paints in WikiArt dataset. The top-3 artists are Vincent van Gogh, Nicholas Roerich, and Pierre Auguste Renoir with 1,889, 1,860, and 1,400 paintings, respectively. The top-10, top-20, and top-30 artists share 14.18%, 21.80%, and 27.62% of the samples, respectively. Figure 3 shows the distribution of the number of paintings and their authors.

We first retrain the CLIP text encoder with the same tokenizer with the LDM fixed. This stage is expected to map the captions used in Wikiart to stable diffusion's latent space. Then, we fine-tune the text encoder and the LDM jointly. This stage is expected to help the LDM to enrich its knowledge of artworks from different artists, in different styles and genres.

5 Experiments

We use a DGX-A100-80GB server with 8 NVIDIA A100-80GB GPU cards. The original code and settings of the stable diffusion model's checkpoint v1-4 is reused. During inferencing, single GPUs are used with ddim_eta=1.0, ddim_steps=200, height and width are both 512, and scale is set to be 5.0.

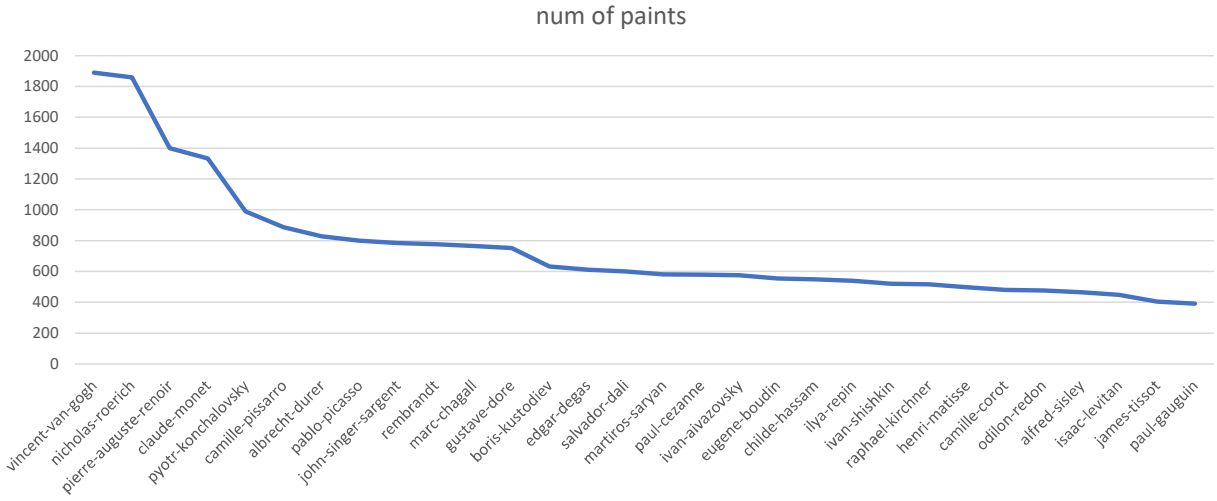

Figure 3: Top-30 artists and their painting numbers in Wikiart.

5.1 Direct Comparison with Original LDMs

Figure 4 directly compares the images generated by the original model and that retrained under Wikiart. We used the same prompts as described in (Rombach et al., 2021). For direct comparison, we also directly copy the first two rows from their original paper. We list four rows picked from the top-30 artists (Figure 3). The painting skills and styles of the artists are reflected. For example, in our first row all "drawn" by Vincent van Gogh, it is relatively easy to distinguish them from other artists: star sky appears often and the Zombie painting is telling a rich story of the author himself.

When the "street sign" is given in the first column, the original paper's two results mainly focused on the photo-style signs themselves. Yet, for the artists, the background's nice street views are also important parts of the final painting, such as the sky, the forest, the building and the people with an orange umbrella. With these hints, we modestly draw a preliminary conclusion that our four paintings (rows 3 to 6) of the first column are more creative and include richer sounding environments and humane information.

Column three, five and six are drawn from prompts which include "fake objects" which do not frequently exist in the real-world. The "half mouse half octopus" is more like photos in the original paper (column 3, first 2 rows), our images are closer to hand-drawn paints. When drawing a "chair that looks like an octopus", all the rows in column six are close to artworks.

The final column can be regarded as an industrial design oriented prompt. With the artists' style and genre included, we can positively imagine that when these paints are printed in real-world T-shirts, people will show their interests of further personalized customization and buy them.

5.2 Textual Condition Extension Results

We use the former example of "urbanization of China" to show the results of textual condition extension. Figure 5 shows four artworks by four famous artists, Vincent van Gogh, Nicholas Roerich, Pierre Auguste Renoir and Claude Monet. Interestingly, the major elements frequently used by artists are also reflected here. For example, the star sky of Vincent van Gogh, the water and boats of Claude Monet. The major elements included in the four paintings are also interesting, combinations of Chinese traditional buildings and skyscrapers, combinations of individual houses and mountains, rather crowded endless buildings and blurry sky, and Chinese traditional building style boats with super high skyscrapers around the rivers.

Figure 6 shows the same four artists' artwork for an extended prompt related to one of the most rapidly developed cities, Shenzhen, during the urbanization of China. With the extended prompts, the model could generate more expressive images. For Vincent van Gogh, a moon in the middle of the sky, with fishing-boats near and high buildings in the far view. The same elements of fish boats and skyscrapers are all included in the other three paintings. Interestingly, for Nicholas Roerich, even the skyscrapers are drawn by following traditional Chinese style.

Figure 7 shows the same four artists' artwork for an extended prompt related to a train running on

the snow-capped mountains, during the urbanization of China. With the extended prompts, again, the model could generate more expressive images and keep the characters of each artist. The general styles and viewpoints of the four artists are reflected: now we have the mountain as the "sky" of Vincent van Gogh and the "sky and mountain" in Claude Monet looks like a reversed river.

Figure 8 shows the same four artists' artwork for an extended prompt related to children running in wheat-fields, during the urbanization of China. With the extended prompts, again, the model could generate more expressive images with rich emotional colors such as blue skies, golden wheat fields, and running-enjoy children. The general styles and viewpoints of the four artists are reflected, such as Vincent van Gogh's sky and the skirts of the two girls from Claude Monet.

Full images of the top-30 artists (Figure 4) of the one initial prompt and three extended prompts are shown in Figure 9, 10, 11 and 12 respectively.

5.3 Diversity and Styles

We finally investigate the diversity and style influences. Figures 13, 14, 15 and 16 shows the 27 styles of Vincent van Gogh, each style with 5 samples (per row), for the former prompt "left-behind children running in wheat-field". Most images are with a "van Gogh" style sky. The diversity is ensured by comparing the columns in each row. Since Vincent van Gogh is famous for "Post Impressionism" (Figure 16, row 2), the characteristics of other styles are relatively less recognizable. The balancing of between keeping the typical style of van Gogh and introducing new styles is relatively difficult. Still, from the five images of style "Ukiyo e" (Figure 14, row 2), we can recognize that the children are with Japanese traditional cloths and hair styles (so do the buildings behind).

6 Conclusion

In order to improve the creativity of LDMs, we have proposed two directions of extending the input prompts and of retraining the original model by the Wikiart dataset. We take the 1,000 artists in recent 400 years as the major source of both creativity and artistry. With these proposals, the resulting diffusion models can ask these famous artists to draw novel and expressive paints of modern topics.

We believe this is an interesting topic and has industrial design requirements for real-world applications, such as cloth designing, advertisement posters, and game character designing. Through drawing the real-world's topics with the help of hundreds to thousands famous artists, it is reasonable to learn the creativity and fertility from these artists' eyes.

References

Hanqun Cao, Cheng Tan, Zhangyang Gao, Guangyong Chen, Pheng-Ann Heng, and Stan Z. Li. 2022. A survey on generative diffusion model.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. 2022. Diffusion models in vision: A survey.

Patrick Esser, Robin Rombach, and Björn Ommer. 2020. Taming transformers for high-resolution image synthesis. *CoRR*, abs/2012.09841.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239.

Abdul Jabbar, Xi Li, and Bourahla Omar. 2020. A survey on generative adversarial networks: Variants, applications, and training. *CoRR*, abs/2006.05132.

Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. 2021. Diff-tts: A denoising diffusion model for text-to-speech.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. 2022. Pseudo numerical methods for diffusion models on manifolds.

Songxiang Liu, Yuewen Cao, Dan Su, and Helen Meng. 2021. Diffsvc: A diffusion probabilistic model for singing voice conversion.

Gautam Mittal, Jesse H. Engel, Curtis Hawthorne, and Ian Simon. 2021. Symbolic music generation with diffusion models. *CoRR*, abs/2103.16091.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail A. Kudinov. 2021. Grad-tts: A diffusion probabilistic model for text-to-speech. *CoRR*, abs/2105.06337.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

S. Robertson. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *CoRR*, abs/2010.02502.

Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wei Ren Tan, Chee Seng Chan, Hernán E. Aguirre, and Kiyoshi Tanaka. 2017. Artgan: Artwork synthesis with conditional categorial gans. *CoRR*, abs/1702.03410.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zhengwei Wang, Qi She, and Tomas Ward. 2021. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys*, 54:1–38.

Julia Wolleb, Florentin Bieder, Robin Sandkühler, and Philippe C. Cattin. 2022. Diffusion models for medical anomaly detection.

Heyang Xue, Xinsheng Wang, Yongmao Zhang, Lei Xie, Pengcheng Zhu, and Mengxiao Bi. 2022. Learn2sing 2.0: Diffusion and mutual information-based target speaker svs by learning from singing teacher.

Ling Yang, Zhilong Zhang, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Ming-Hsuan Yang, and Bin Cui. 2022. Diffusion models: A comprehensive survey of methods and applications.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *CoRR*, abs/1911.00536.

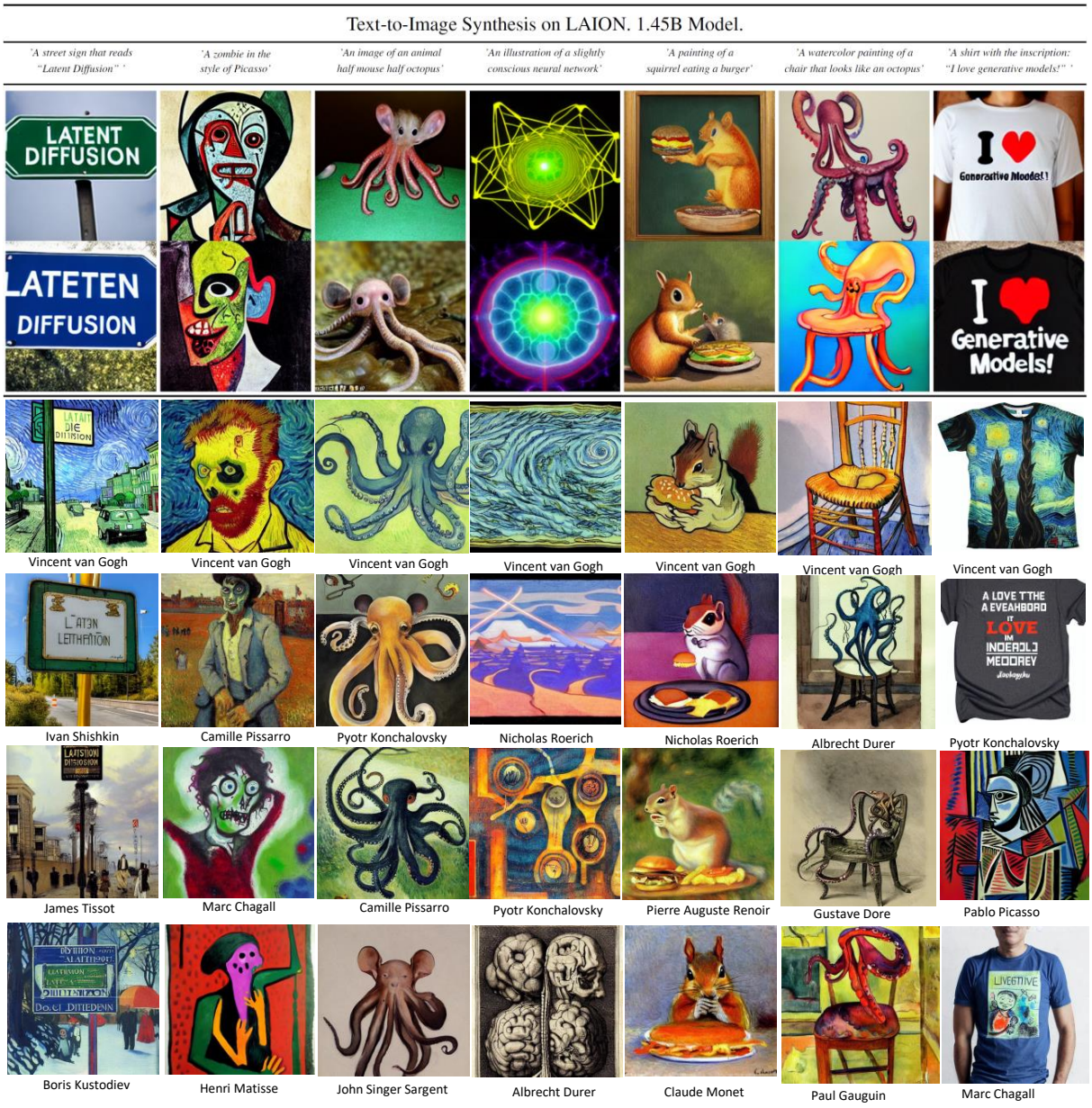

Figure 4: Direct comparison with the same prompts used in (Rombach et al., 2021) yet different artists.

Figure 5: Four artists' artworks for the same prompt of "a painting of urbanization of china".

Figure 6: Four artists' artworks for the same extended prompt of "originally a collection of fishing villages, Shenzhen rapidly grew to be one of the largest cities in China".

Figure 7: Four artists' artworks for the same extended prompt of "a train runs on the snow-capped mountains of the Qinghai-Tibet Plateau".

Figure 8: Four artists' artworks for the same extended prompt of "left-behind children running in wheat-field".

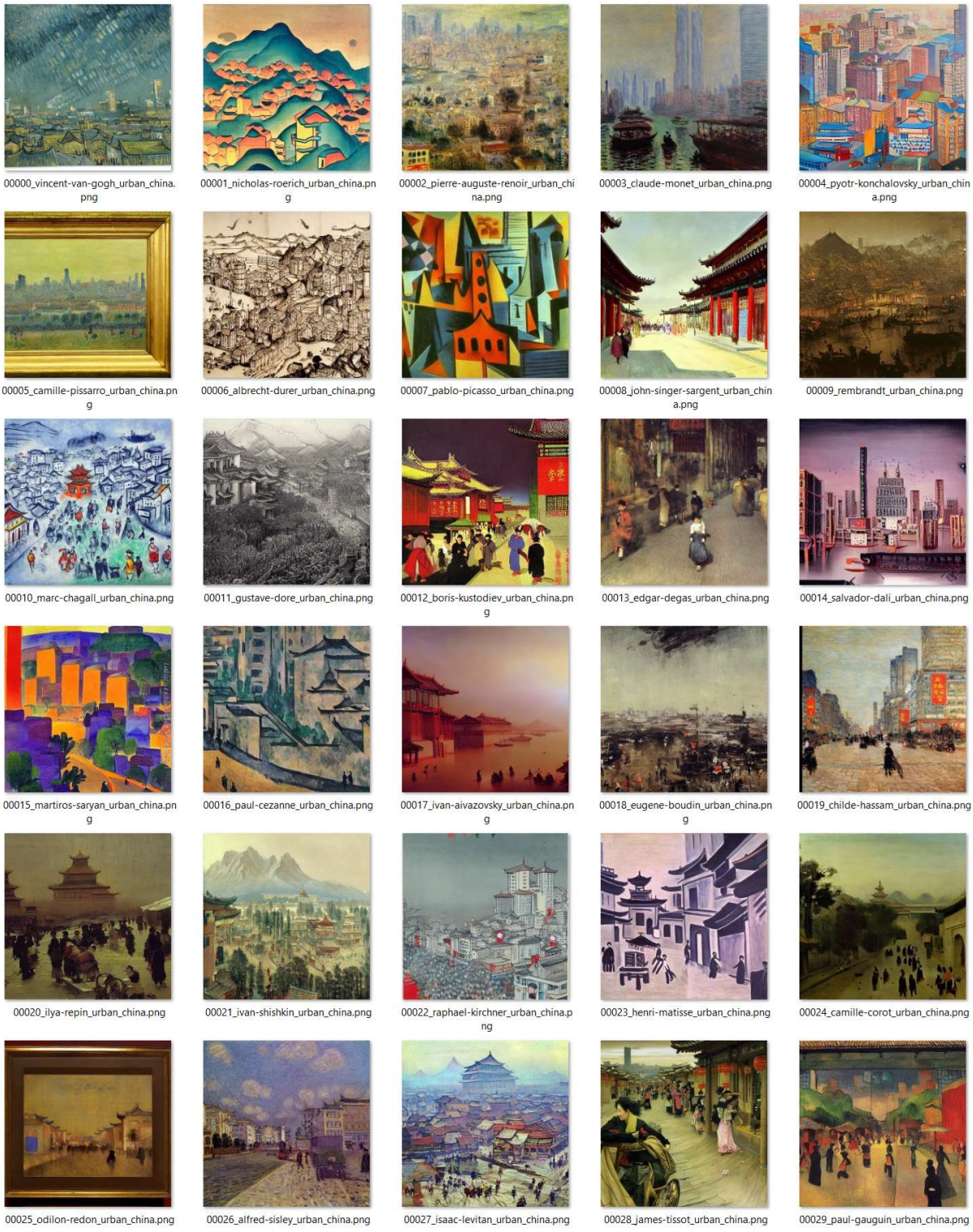

Figure 9: Top-30 artists' artworks for the same extended prompt of "a painting of urbanization of china".

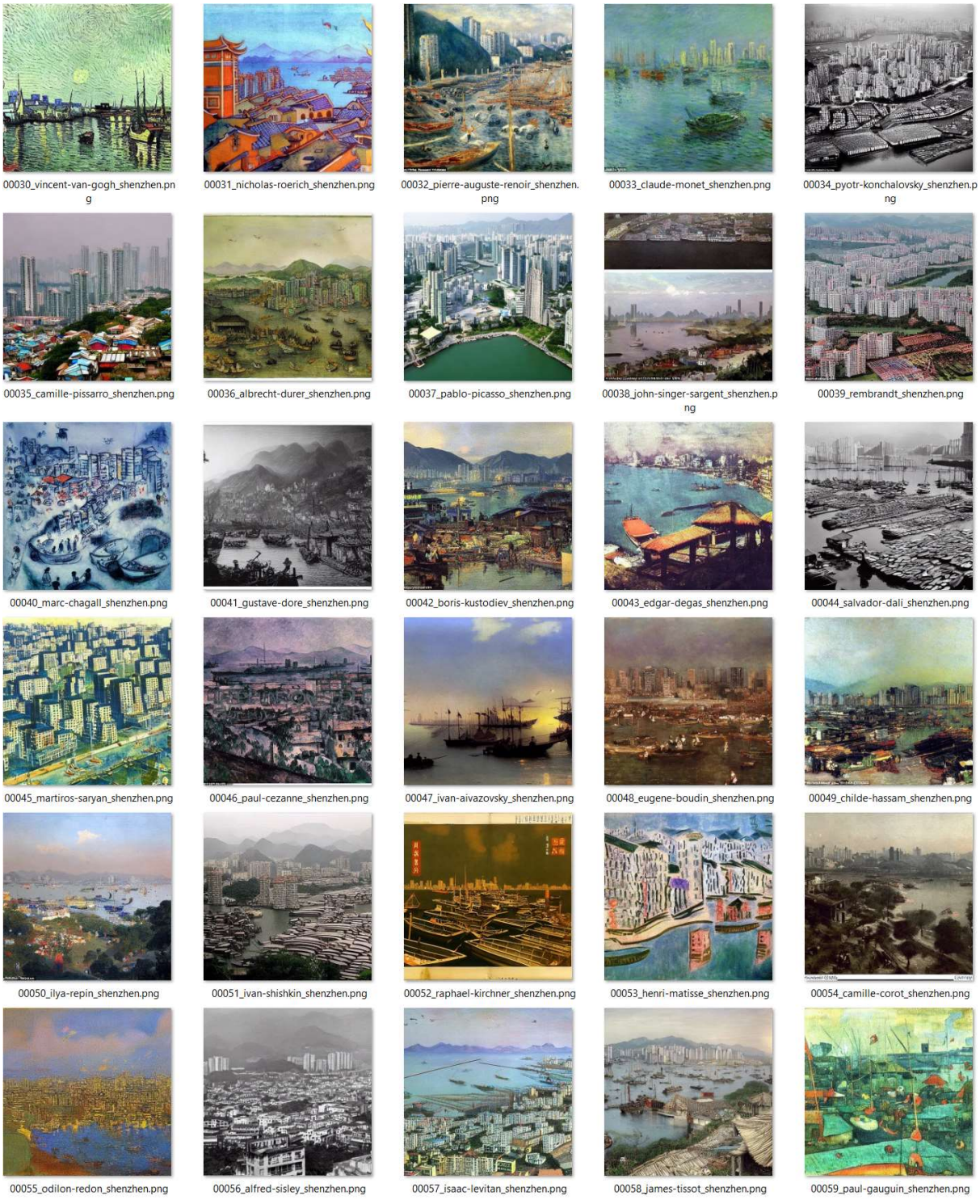

Figure 10: Top-30 artists' artworks for the same extended prompt of "originally a collection of fishing villages, Shenzhen rapidly grew to be one of the largest cities in China".

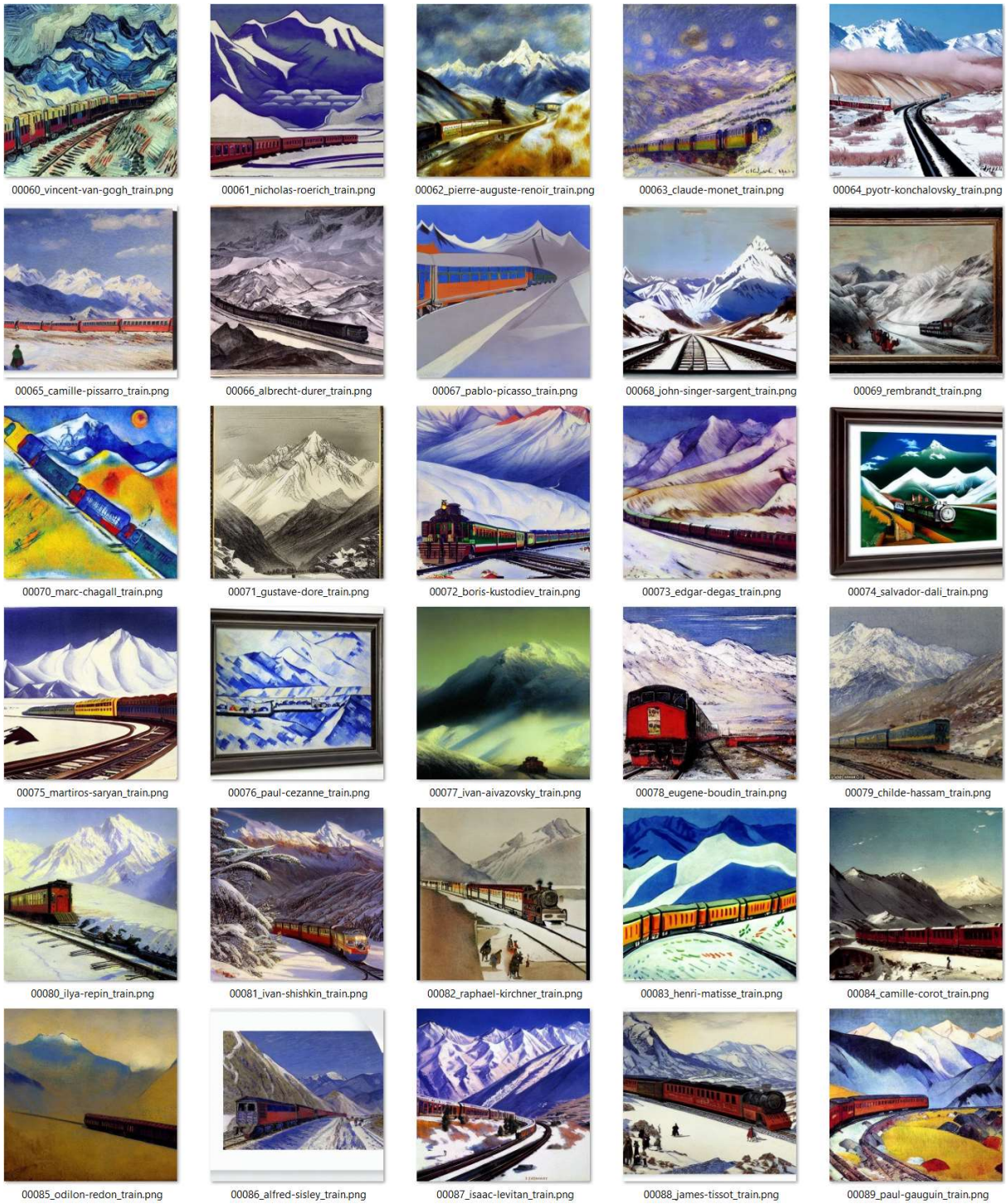

Figure 11: Top-30 artists' artworks for the same extended prompt of "a train runs on the snow-capped mountains of the Qinghai-Tibet Plateau".

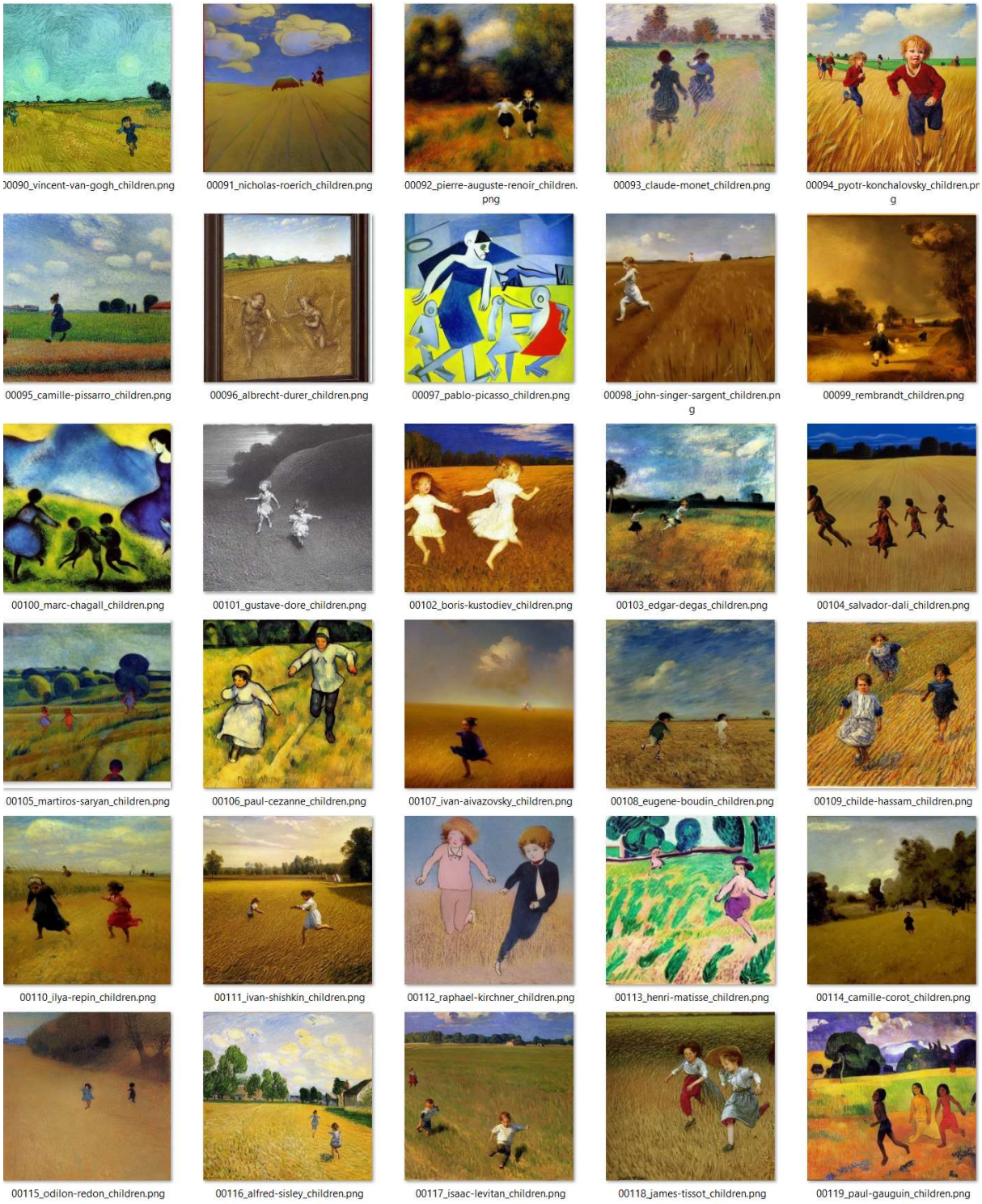

Figure 12: Top-30 artists' artworks for the same extended prompt of "left-behind children running in wheat-field".

Figure 13: Vincent van Gogh's seven styles (Minimalism, Abstract Expressionism Fauvism, Naive Art Primitivism, Symbolism, Color Field Painting, Pointillism), each style with five samples (per row).

Figure 14: Vincent van Gogh's seven styles (Baroque, Ukiyo e, Early Renaissance, Action painting, Contemporary Realism, Mannerism Late Renaissance, Analytical Cubism), each style with five samples (per row).

Figure 15: Vincent van Gogh's seven styles (New Realism, Northern Renaissance, Cubism Impressionism, Expressionism, Realism, High Renaissance), each style with five samples (per row).

Figure 16: Vincent van Gogh's six styles (Pop Art, Post Impressionism, Synthetic Cubism Art Nouveau Modern, Rococo, Romanticism,), each style with five samples (per row).