Inverse Painting: Reconstructing The Painting Process

BOWEI CHEN, University of Washington, USA
YIFAN WANG, University of Washington, USA
BRIAN CURLESS, University of Washington, USA
IRA KEMELMACHER-SHLIZERMAN, University of Washington, USA
STEVEN M. SEITZ, University of Washington, USA

A IMPLEMENTATION DETAILS

In this section, we present the implementation details.

A.1 One-Step Canvas Rendering Approach

Please refer to Sec. A.3 for implementation details.

A.2 Training: Instruction Generation

For text instruction generator g_{text}, we set the question prompt p to: "There are two images side by side. The left image is an intermediate stage in a painting process of the right image. Please tell me what content should be painted next? The answer should be less than 2 words."

We obtain the ground-truth text instructions p_t with the assistance of the pretrained LLaVA 1.5 model "LLaVA-v1.5-7B[1]". Specifically, we horizontally concatenate the ground-truth current image I_{t-1} with the ground-truth next image I_t. We then input this concatenated image alongside the modified question prompt p', which reads: "There are two images side by side. The right image is the next step of the left image in the painting process of a painting. Please tell me what is added to right image? The answer should be less than 2 words." The LLaVA model then outputs the text instructions p_t, where we manually correct any inaccuracies in p_t.

For fine-tuning g_{text}, we employ LoRA and train for 10 epochs, using a learning rate of 1e-4 and a batch size of 16. The fine-tuning process takes approximately 5 hours on a single NVIDIA A100 GPU.

For the mask generator g_{mask}, we implement this as the UNet architecture proposed by Stable Diffusion [Rombach et al. 2022]. The input layer of the UNet is modified to accept a 9-channel input, tailored for the spatial input components $[E_I(I_T), E_I(I_{t-1}), M_d]$. For the time encoder g_t, we implement 3 fully connected layers that progressively increase the input feature dimension from 21 to 256, 512, and finally 768. A ReLU activation function follows each fully connected layer, with the exception of the last one to allow for linear output transformation. We train g_{mask} and g_t for 80k steps with a learning rate of 1e-5 and a batch size of 1. The training process takes approximately 13 hours on a single NVIDIA A100 GPU.

A.3 Training: Canvas Rendering

The feature extractor g_f takes as input the ground-truth current image I_{t-1} and region mask M_t, outputs an encoded feature of them. This feature extractor is implemented as a shallow network containing 9 convolutional layers, which scales the input spatial resolution by 8. The next CLIP generator, denoted as g_c, accepts the CLIP embeddings of both the ground-truth current image, $CLIP(I_{t-1})$, and the target image, $CLIP(I_T)$, as inputs. It then outputs a prediction for the CLIP embedding of the next image. Within g_c, the embeddings $CLIP(I_{t-1})$ and $CLIP(I_T)$ are concatenated along the feature dimension. This concatenated vector is subsequently processed through a three-layer multi-layer perceptron (MLP). The MLP's layers map the features from dimensions of 1536 to 768, 384, and back to 768 respectively. The ReLU activation function is employed at each layer except the final one, where no activation is applied. The time encoder g_t consists of 3 fully connected layers, same as that introduced in Sec.A.2. For more details on the implementation of ReferenceNet g_r, please refer to [Hu et al. 2023a].

We initialize g_u using RealisticVision V5.1 [SG_161222 2023]. The models within canvas rendering, namely g_u, g_r, g_f, g_t, and g_c, are jointly trained using a learning rate of 1×10^{-5} and a batch size of 1. Each conditional signal is dropped (set to zero) 10 percent of the time, in accordance with the classifier-free guidance method [Ho and Salimans 2022]. This enables us to control the strength of each conditional signal at the test time inference. We train the models in 200k steps, taking around 34 hours on a single NVIDIA A100 GPU.

For the one-stage approach outlined in Sec. 3.1 of the main paper, we use the same training strategy but exclude text and mask instructions.

A.4 Test-Time Generation

At a specific step $t-1$, we render the subsequent image $\hat{I}_t$ using the trained pipeline. The denoising process of the diffusion renderer employs a scheduler based on ancestral sampling, specifically utilizing the Euler method steps [Karras et al. 2022]. The denoising timestep S is set to 25. For classifier-free guidance, we assign guidance scales of 5 for text, mask, and time interval, and a scale of 2 for the next CLIP embeddings. Each update in this process takes approximately 4 seconds on a single NVIDIA A100 GPU. The generation process is halted if the perceptual distances between $\hat{I}_{t-2}$ and $\hat{I}_{t-1}$, and between $\hat{I}_{t-1}$ and $\hat{I}_t$, are both less than 1×10^{-3}.

A.5 Baselines Details

For *Timecraft*, we train the model on our dataset using the default settings provided in the official code. The training consists of two stages: pairwise optimization and sequence optimization. In the first stage, we train the model for 500K steps, which takes approximately 4 hours on 2 TITAN XP GPUs. In the second stage, we train the model for 78K steps, which takes around 25 hours on 2 TITAN XP

[1] https://huggingface.co/liuhaotian/llava-v1.5-7b

Authors' addresses: Bowei Chen, University of Washington, 1410 NE Campus Pkwy, Seattle, WA, 98195, USA, boweiche@cs.washington.edu; Yifan Wang, University of Washington, Seattle, USA, yifan1@cs.washington.edu; Brian Curless, University of Washington, Seattle, USA, curless@cs.washington.edu; Ira Kemelmacher-Shlizerman, University of Washington, Seattle, USA, kemelmi@cs.washington.edu; Steven M. Seitz, University of Washington, Seattle, USA, seitz@cs.washington.edu.

GPUs. We observed that training with more steps will degrade the model performance.

For *Stable Video Diffusion (SVD)*, we fine-tune a 14-frame model on our dataset using LoRA [Hu et al. 2021]. During fine-tuning, we sample one frame from our training sequences as input and use its previous 13 frames as ground truth, padding with white images when necessary. The target image is used as the input frame 40% of the time, while other images are randomly selected otherwise. We fine-tune the model for 2K steps, which takes around 1.5 hours on 4 NVIDIA A100 GPUs. Fine-tuning for 2K steps yields the best performance; more steps cause the model to produce painting videos that get stuck, while fewer steps result in underfitting, causing the camera viewpoint to shift.

For *Paint Transformer*, we use the pretrained model provided by the authors for our comparisons. This model generates 200 frames given an input painting. We additionally evaluate two stroke-based rendering baselines [Hu et al. 2023b; Zou et al. 2021] and an amodal segmentation baseline [Ozguroglu et al. 2024] in Sec. B, please refer to Sec. B.2 for their implementation details.

B EXPERIMENTS

B.1 More Results

In Fig. 5 and Fig. 6, we show more results of our method. As discussed in the main paper, our method can handle paintings with different styles and generate human-like painting process in terms of painting order, focal region and layering techniques.

B.2 More Baselines

We compare our method with two additional stroke-based rendering baselines and an amodal segmentation baseline.

Stylized Neural Painting [Zou et al. 2021] employs an optimization-based approach featuring a novel neural renderer that mimics vector renderer behavior. Here, the stroke prediction is framed as a parameter search process aiming to maximize the similarity between the input image and the rendered output. We utilize the pretrained network "the oil-paint brush" provided by the authors for comparison. This method generates 499 frames given a target painting.

Compositional Neural Painter [Hu et al. 2023b] utilizes a phased RL strategy for predicting paint regions and a painter network to determine stroke parameters. A neural stroke renderer is then trained to apply the strokes onto the canvas based on the predicted stroke parameters. We use the authors' pretrained networks for comparison. This method generates 50 frames from a target painting.

pix2gestalt [Ozguroglu et al. 2024] completes a partially visible object in the image given the partial segment mask of the object. We adapt it to out task as follows: (1) segment the target image using [Liu et al. 2023], (2) complete each segment using the pretrained model of *pix2gestalt*, (3) place completed segments on the canvas by depth (farther first). The depth of each segment is determined by the average depth of its pixels, where the depth is estimated using a pretrained depth estimation model [Yang et al. 2024]. Please note that we define the painting order heuristically, as the baseline does not support learning this order.

Similar to the strategy used for *Paint Transformer* in the main paper, we set the time intervals for these three baselines based on the average training video duration (561 seconds) divided by the number of frames. This results in time intervals of 1.12 seconds for *Stylized Neural Painting* and 11.22 seconds for *Compositional Neural Painter*. The time interval of *pix2gestalt* varies for different target images, depending on the number of detected segments in the target image.

Evaluation on Full Paintings						
Method	LPIPS ↓	IoU ↑	DDC ↓	DTS ↓	FID ↓	FVD ↓
Stylized Neural Painting	0.669	0.031	122.2	8.782	358.3	1457
Compositional Neural Painter	0.680	0.049	91.93	7.507	374.8	1505
pix2gestalt	0.609	0.214	126.3	9.089	341.9	1576
Paint Transformer	0.643	0.104	94.61	6.057	337.3	1616
Timecraft	0.602	0.251	153.2	9.964	289.8	1582
SVD	0.500	0.197	135.5	8.577	168.3	1594
Ours-TE-TG-MG	0.468	0.128	88.13	6.204	203.3	1591
Ours-TG-MG	0.447	0.139	62.79	4.913	187.4	1468
Ours-TE	0.413	0.375	58.81	4.153	167.5	1319
Ours-MG	0.435	0.175	61.09	3.972	182.5	1471
Ours-TG	0.399	0.400	39.41	2.120	161.1	1418
Ours-RN	0.416	0.396	46.71	1.542	174.2	1464
Ours-CE	0.371	0.402	34.16	1.346	158.4	1326
Ours 10	0.369	0.349	35.27	1.693	158.7	1347
Ours 30	0.387	0.353	36.26	1.933	151.7	1279
Ours	**0.364**	**0.418**	**32.66**	**1.273**	**150.6**	**1273**
Evaluation on Cropped Paintings						
Timecraft	0.647	0.165	166.29	6.743	363.0	1627
Ours	**0.452**	**0.296**	**56.62**	**2.545**	**197.2**	**1034**

Table 1. **Comparison with baselines and our ablation variants on the full and cropped paintings.** Our full model (with a time interval of 20) outperforms all of them.

B.3 More Metrics

We also evaluate the quality of the generated videos using the Fréchet Video Distance (FVD) [Ge et al. 2024]. While FVD might not be ideally suited for assessing time-lapse painting process videos, we include it for the sake of comprehensiveness.

B.4 Baseline Comparison

We present more comparisons with baseline methods on both full and cropped paintings.

Full Paintings. We provide qualitative comparisons with all baselines for full paintings in Fig. 9, Fig. 10, Fig. 11, Fig. 12, and Fig. 13. The three stroke-based rendering baselines – *Stylized Neural Painting*, *Compositional Neural Painter*, and *Paint Transformer* – apply brushstrokes in a non-human-like manner, as they are not trained on actual painting videos. Furthermore, due to the limitations of parameterized brushstroke constraints, they produce only a "stylized" version of the target painting. *pix2gestalt*'s results are non-human-like and exhibited visual artifacts. This is due to inaccurate predefined painting order, imperfect segmentation, and unnecessary paints on the canvas to complete unoccluded segments. *SVD* often gets stuck in the painting process, evident in columns 4 to 6 of Fig. 12, and produces visual artifacts, such as the unreasonable colors in columns 1 and 2 of Fig. 10 and columns 1 to 5 of Fig. 11. Moreover, despite being trained on a real painting dataset, it still fails to mimic the human painting order reasonably. *Timecraft* generates only very low-resolution sequences and introduces noticeable visual artifacts. In contrast, our method significantly outperforms all baselines in mimicking human-like painting sequences, focusing on focal areas, employing layering techniques, and achieving good video quality.

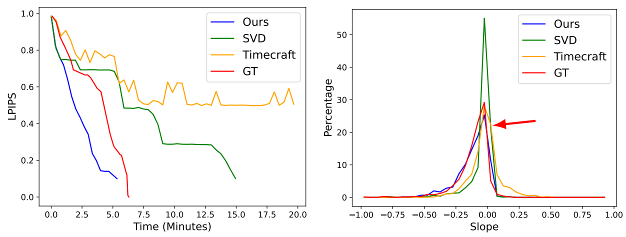
Table. 1 presents the quantitative comparisons across all baselines and metrics. Our method outperforms all baselines in every evaluated metric.

Further, we evaluate how the painting processes generated by various methods progress toward the target painting in terms of speed and direction using the distance curve. The distance curve of a sequence depicts its progression towards the target painting, plotting time (x-axis, minutes) against LPIPS distance (y-axis) between target and current images. Among baselines, we select *SVD* and *Timecraft* for analysis. For these two baselines, we use a time interval of 23.6 seconds, matching the average duration of our training set. Fig.1 (a) illustrates the distance curve of various methods applied to a single painting in the validation set. *Timecraft* fluctuates considerably and fails to consistently approach the target image. Besides, it does not converge because its outputs are in a very low resolution. *SVD* encounters stalls during the painting process, leading to extended convergence times. Our method exhibits the curve that most closely aligns with that of the GT. Note that, neither our method nor *SVD* achieves perfect reconstruction of the target painting due to the utilization of VAE. For further analysis, Fig.1 (b) presents the distribution of the slope of these distance curves across all paintings in the validation set. *Timecraft* exhibits over 20% of its slopes between 0 to 0.05 (marked with a red arrow), indicating frequent deviations from the target image, such as erasing and adding unrelated colors. *SVD*'s slopes, with over 50% ranging from -0.05 to 0, indicate minimal updates on the canvas. In contrast, our method's distributions are closer to those of the GT, demonstrating that our painting process progresses at a reasonable speed and direction.

Cropped Paintings. We compare with *Timecraft* by randomly selecting 3 low-resolution crops from every downsampled full painting in the validation set, following the cropping strategy presented in the paper of *Timecraft*. Fig. 2 provides a qualitative comparison of cropped paintings with *Timecraft*, where our approach yields a more authentic painting process in terms of painting order, focus regions, and overall video quality. The quantitative results in Table. 1 further demonstrate that our method outperforms *Timecraft* across all metrics.

B.5 Ablation Study

In Fig. 7, we present the ablation studies for variants omitting different conditional signals. Both one-step variants, *Ours-TE-TG-MG* and *Ours-TG-MG*, yield unsatisfactory results, underscoring the significance of the two-stage design. *Ours-MG* heavily depends on text instructions and tends to complete an entire semantic class with each update, unexpectedly accelerating the generation process excessively. *Ours-TG* struggles to comprehend the semantic contents of the target paintings, consequently painting the grass and flowers simultaneously without employing layering techniques. We further present qualitative results of the variants without considering predicted CLIP embedding (*Ours-CE*) in Fig. 3. It completes details of the mountain at an early stage, which fails to mimic the painting style of artists in our training set. In contrast, *Ours* delivers the most human-like painting process, characterized by a logical painting order, targeted focal regions, and proper use of layering techniques.



(a) DC of a Single Painting (b) Slope Distribution of DC

Fig. 1. **Quantitative analysis of painting process.** (a) illustrates the distance curves (DC) of various methods applied to a specific painting; (b) visualizes the distribution of slope of these distance curves across all paintings in the validation set. In (b), we classify the slopes into 20 evenly spaced intervals ranging from -1 to 1, and plot the percentage of the slope value (y-axis) falls into each interval (x-axis). For instance, the peak of the green curve in (b) shows that over 50 percent of the slopes range from -0.05 to 0. A negative slope value suggests that the update to the canvas bring it closer to the target image (as expected), and vice versa.

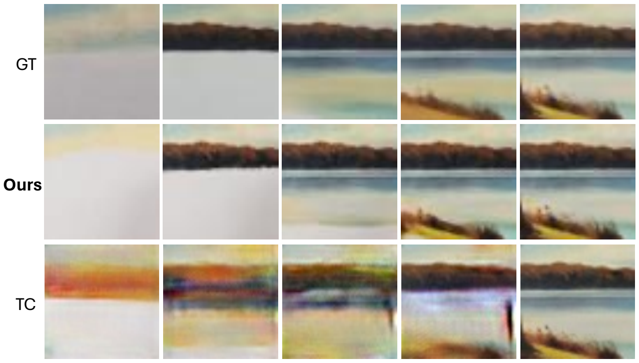


Fig. 2. **Qualitative comparison on low-resolution painting crops.** We compare with *Timecraft* (TC) on low-resolution painting crops. *Timecraft* produces artifacts and fails to produce a human-like painting process. In contrast, our method delivers a more realistic painting video with better quality.

B.6 Human Study

As described in the main paper, we normalized the ratings to remove user bias. Specifically, we divided each participant's rating for a specific painting sequence by the sum of their ratings for all 4 painting sequences (of the same target painting). We then averaged these normalized ratings across all paintings and participants for each method.

Our method achieved the highest average normalized rating, surpassing *SVD* by 1.9 times, *Paint Transformer* by 2.1 times, and *Timecraft* by 3.0 times (reported in the main paper). Without normalization, our method received the highest rating at 4.21, compared to *SVD* (2.96), *Paint Transformer* (2.11), and *Timecraft* (1.52).

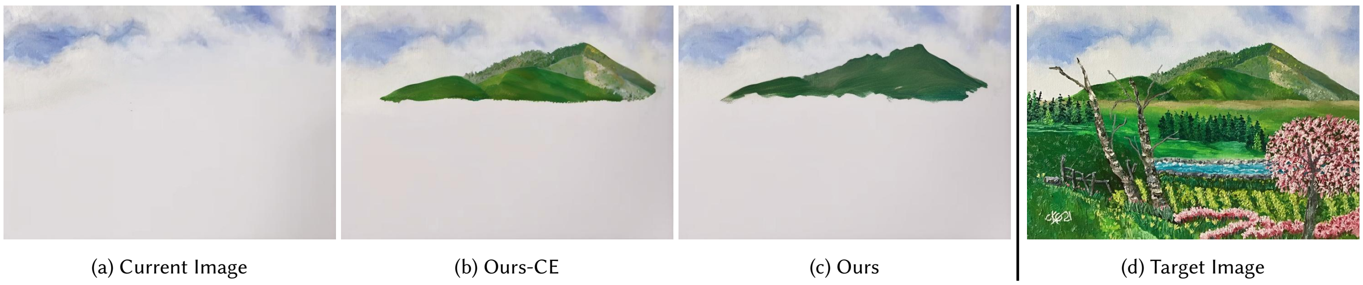


Fig. 3. **Ablation of the predicted CLIP embeddings.** The current and target images are shown in (a) and (d), respectively. (b) and (c) represent the outputs generated by different models at the same timestep, utilizing the same time interval, current image, and target image. Without incorporating predicted CLIP embeddings (b), the model completes the mountain in full detail. The process involves frequent switching of color brushes, deviating from the painting style observed in the training set. Our full model (c) paints the base layer of the mountain first and leaves the details for latter stages, which follows the artistic techniques presented by the artists in our training data. Please see Fig. 6 in the main paper for more keyframes generated by the full model. Image courtesy Catherine Kay Greenup.

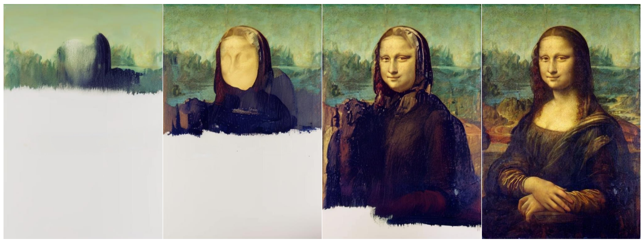


Fig. 4. **More failure cases.** Trained on landscape paintings, our method struggles with portrait paintings. Image courtesy Rawpixel.

B.7 Influence of Random Seeds

In Fig. 8, we illustrate the outcomes of utilizing different random seeds for the diffusion model in our methods. Although different seeds are used, the generated painting processes generally follow a similar order, with minor variations in the sequence of painting foreground objects. These variations are reflective of those observed in the training data, demonstrating that our method can learn the general painting order from these slightly varied sequences and capture the variability. This adaptability allows for the use of different random seeds at inference time to achieve diverse results.

B.8 Failure Case

Fig. 4 shows another failure case of our method on the portrait painting, Mona Lisa.

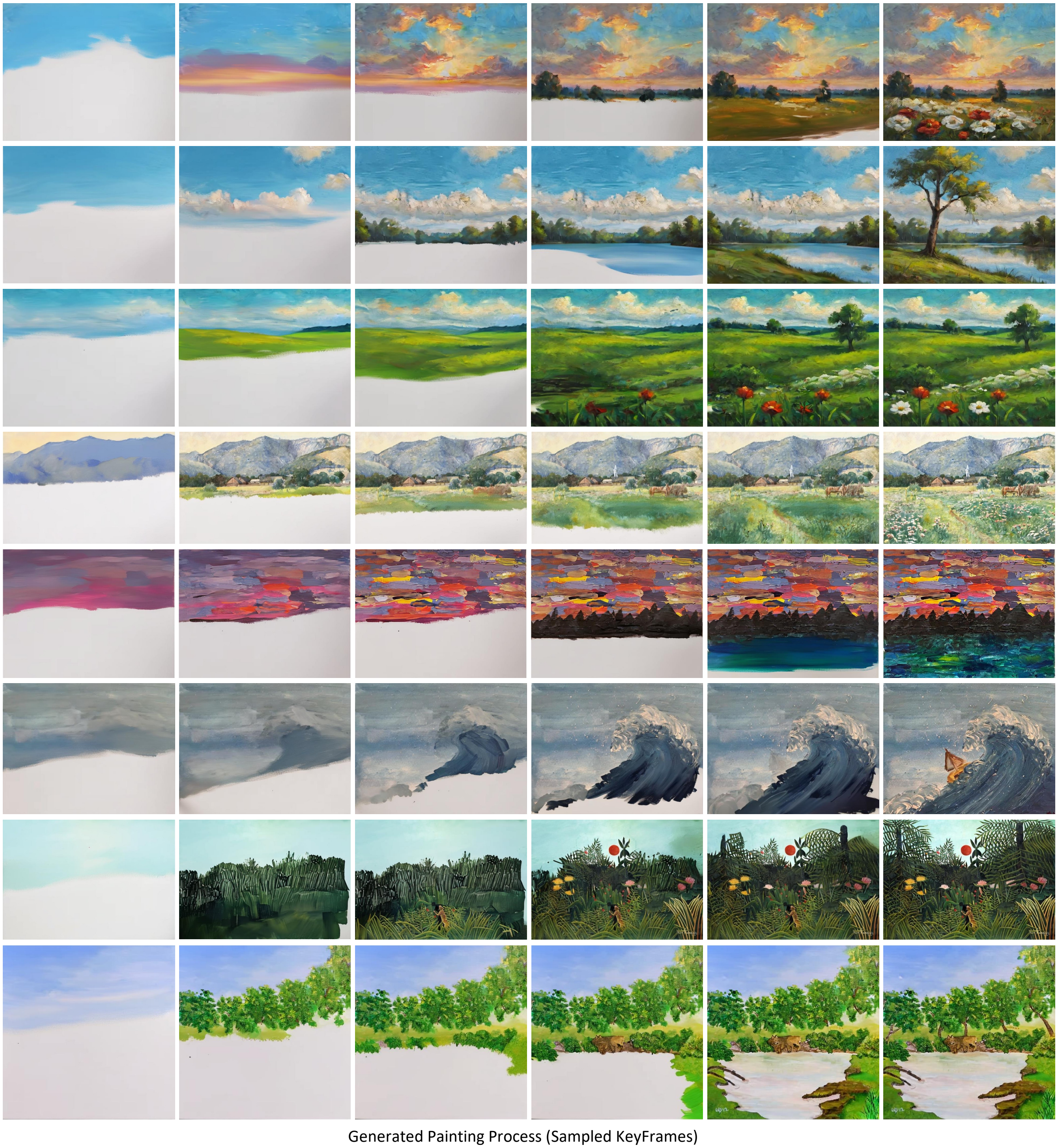


Fig. 5. **More qualitative results on our method.** We show our results on in-the-wild paintings, where the left 5 columns are sampled frames from the generated painting process, and the rightmost column is the target image. Our method effectively handles paintings across various artistic styles, color themes, and aspect ratios. The generated sequences showcase human-like painting orders, maintain reasonable focal regions during different phases, and employ layering painting techniques. Images courtesy Michelle Shlizerman and Catherine Kay Greenup. Image courtesy Julius Zorkoczy, Landscape with a Blooming Meadow, Slovenska narodna galeria, SNG, https://www.webumenia.sk/dielo/SVK:SNG.K_1710. Image courtesy Henri Rousseau, Virgin Forest with Sunset, Kunstmuseum Basel Museum.

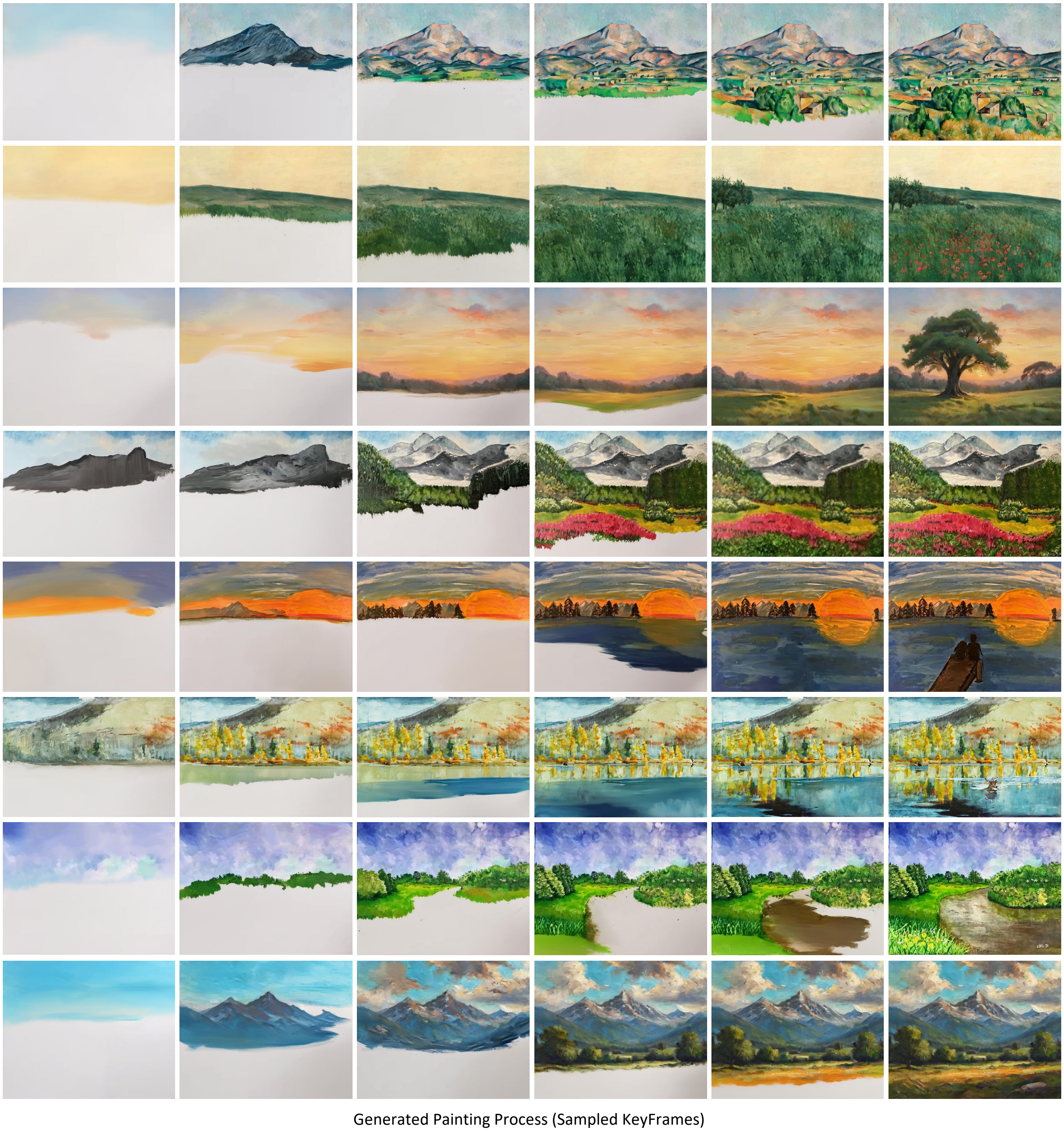


Fig. 6. **More qualitative results on our method.** We show our results on in-the-wild paintings, where the left 5 columns are sampled frames from the generated painting process, and the rightmost column is the target image. Our method effectively handles paintings across various artistic styles, color themes, and aspect ratios. The generated sequences showcase human-like painting orders, maintain reasonable focal regions during different phases, and employ layering painting techniques. Images courtesy Barnes Foundation, Catherine Kay Greenup, The Clark Art Institute and Michelle Shlizerman. Image courtesy František Kaván – Red Poppies, 1910, Slovenská národná galéria, SNG, https://www.webumenia.sk/dielo/SVK:SNG.O_4549.

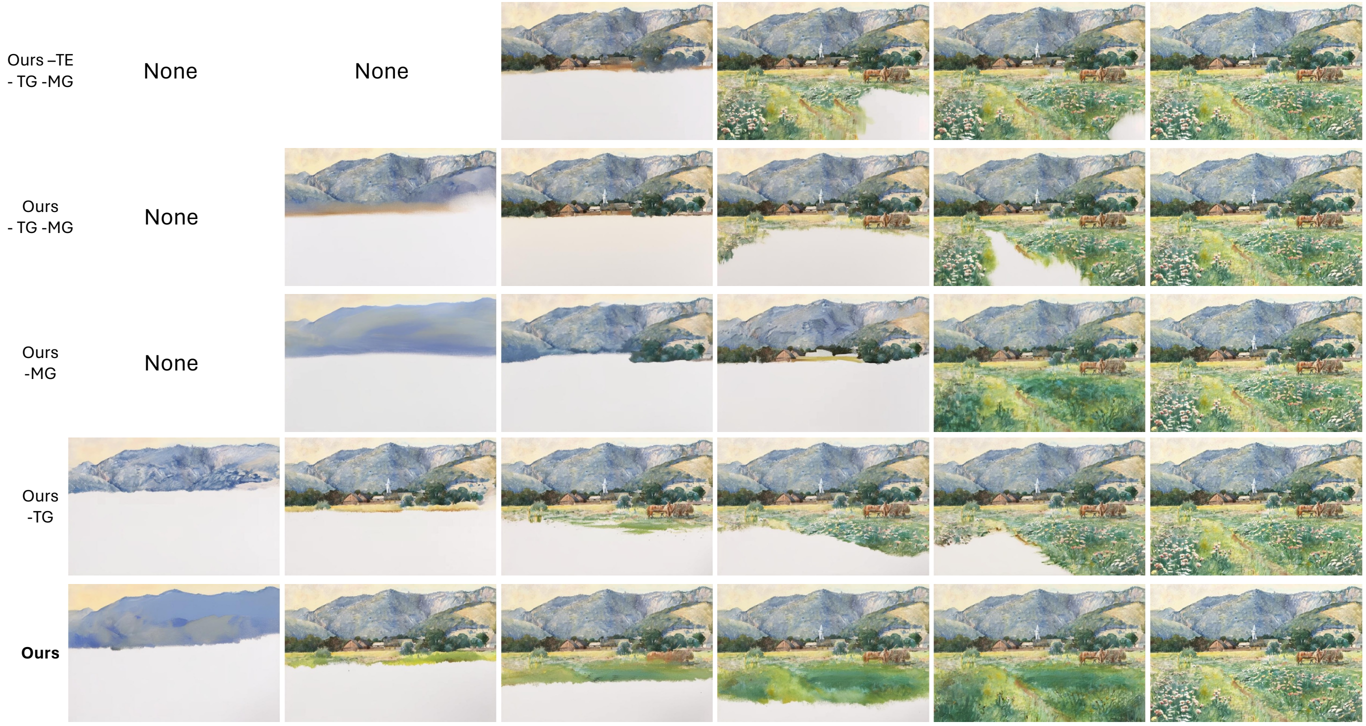


Fig. 7. **Ablation study of different conditional signals.** The row with "None" means there are no enough keyframes for display. For instance, the first row has only four keyframes, leaving the first two columns "None". *Ours-TE-TG-MG* generates excessive content for each update, resulting in only four keyframes throughout the entire painting process. *Ours-TG-MG* slows generation slightly but still converges quickly. Besides, it also produces unreasonable rendering, as shown in column 2. *Ours-TG*, relying heavily on text instructions, tends to complete entire semantic classes in each update, promoting swift convergence. *Ours-MG* leverages region masks to moderate the speed of generation. Yet, in the absence of textual guidance, it struggles to adequately differentiate between semantic classes. For example, in column 4 to 6, the flower and grass are painted simultaneously, rather than using layering techniques that complete the grass first and add the flower afterward. In contrast, *Ours* achieves a more logical and orderly painting process. Image courtesy Julius Zorkoczy, Landscape with a Blooming Meadow, Slovenska narodna galeria, SNG, https://www.webumenia.sk/dielo/SVK:SNG.K_1710.

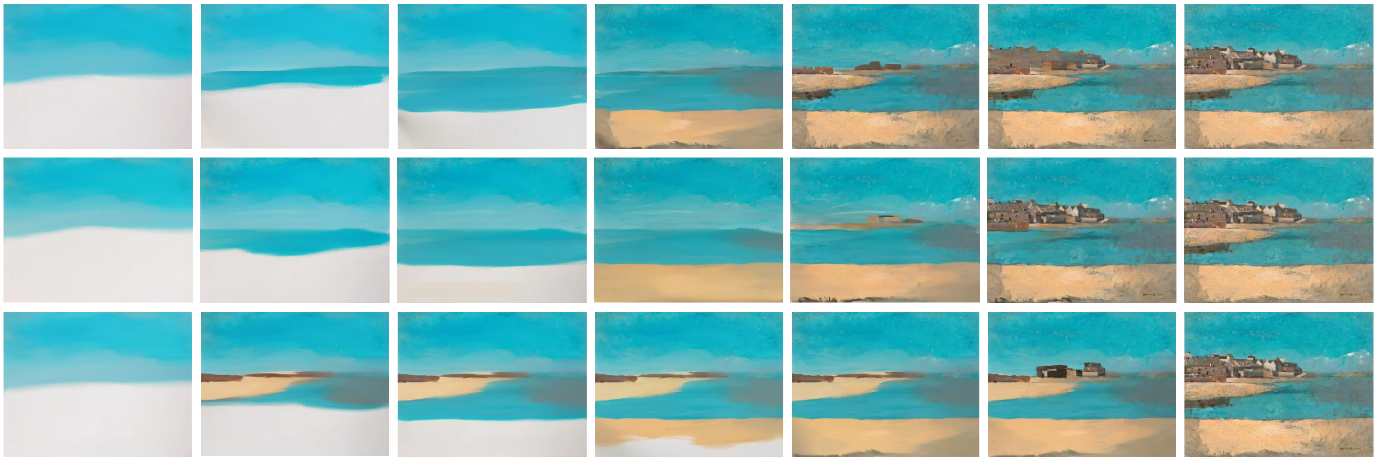


Fig. 8. **Comparison of using different random seeds for the diffusion model in our method.** Each row displays the results of a specific random seed. Despite using different random seeds, the painting orders are generally similar (i.e., back to front) with slight differences in painting foreground objects. For example, rows 1 and 2 tend to address the far ground and house in the final stages, whereas row 3 completes the far ground immediately after finishing the background nearby. Both styles of painting order are observed in the training set, and our method successfully captures these variations. Image courtesy National Gallery of Art.

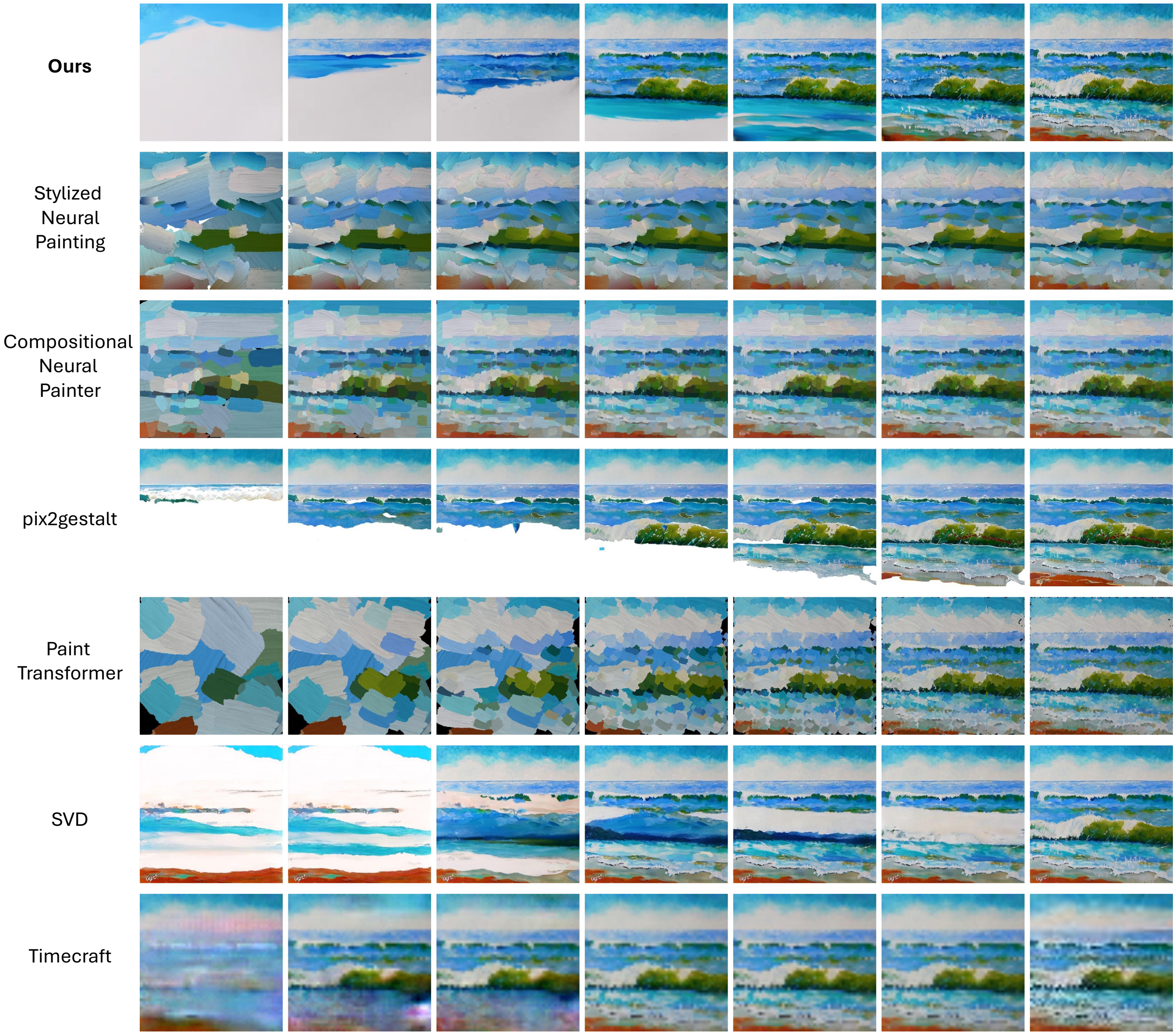


Fig. 9. **Comparison with baselines on in-the-wild images.** Our method significantly outperforms these baselines in generating a more human-like painting video with better visual quality. Image courtesy Catherine Kay Greenup.

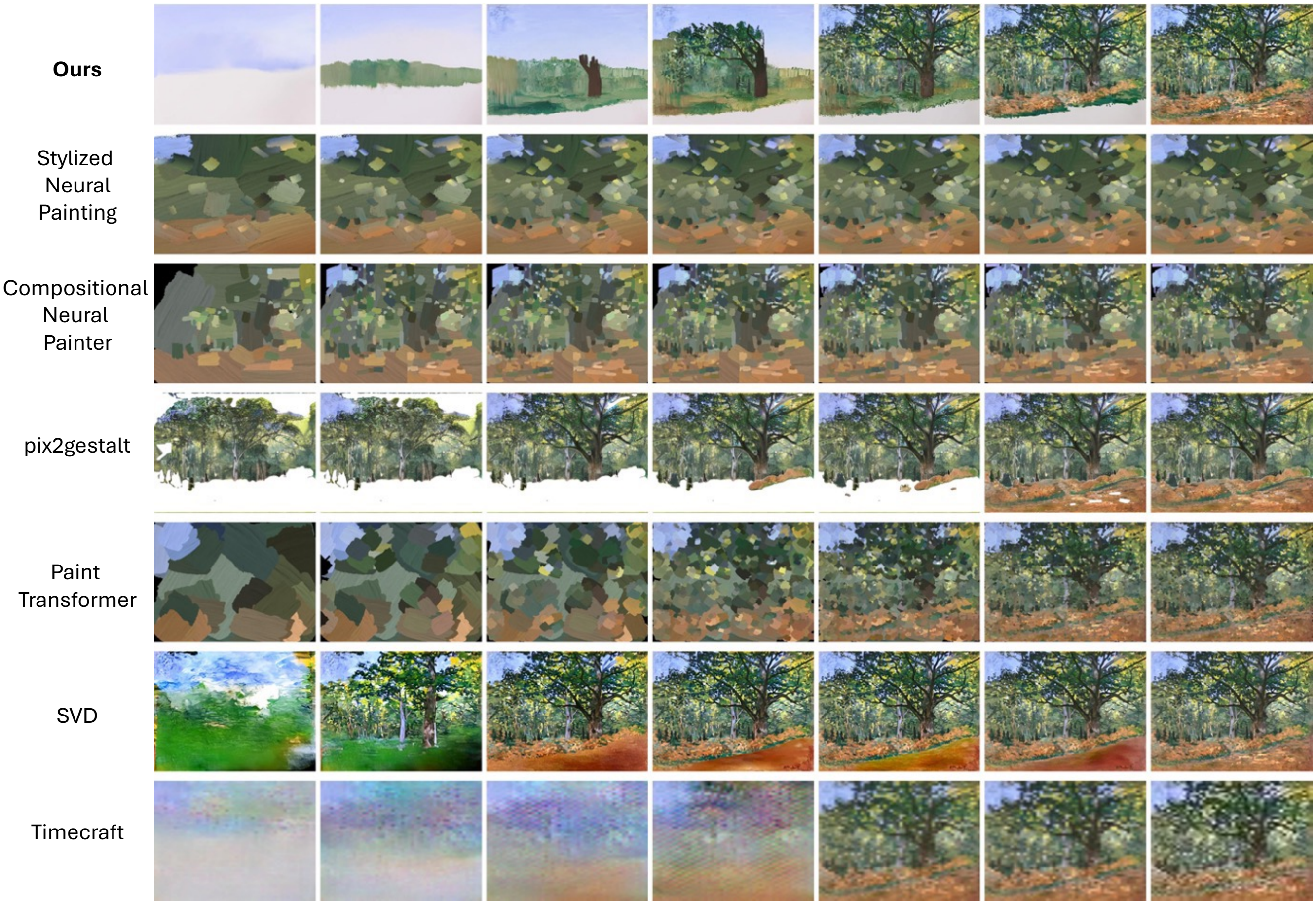


Fig. 10. **Comparison with baselines on in-the-wild images.** Our method significantly outperforms these baselines in generating a more human-like painting video with better visual quality. Image courtesy Rawpixel.

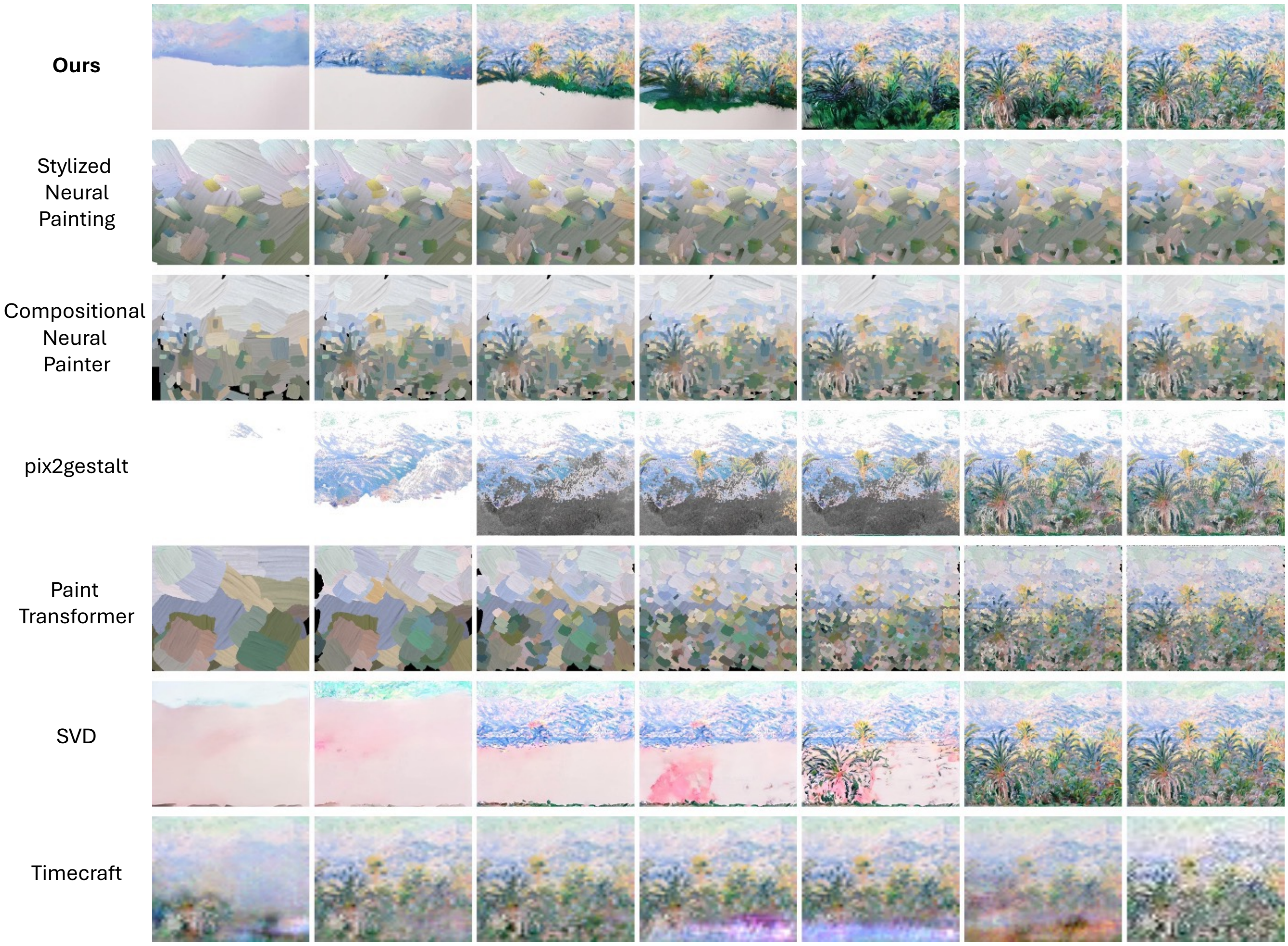


Fig. 11. **Comparison with baselines on in-the-wild images.** Our method significantly outperforms these baselines in generating a more human-like painting video with better visual quality. Image courtesy Rawpixel.

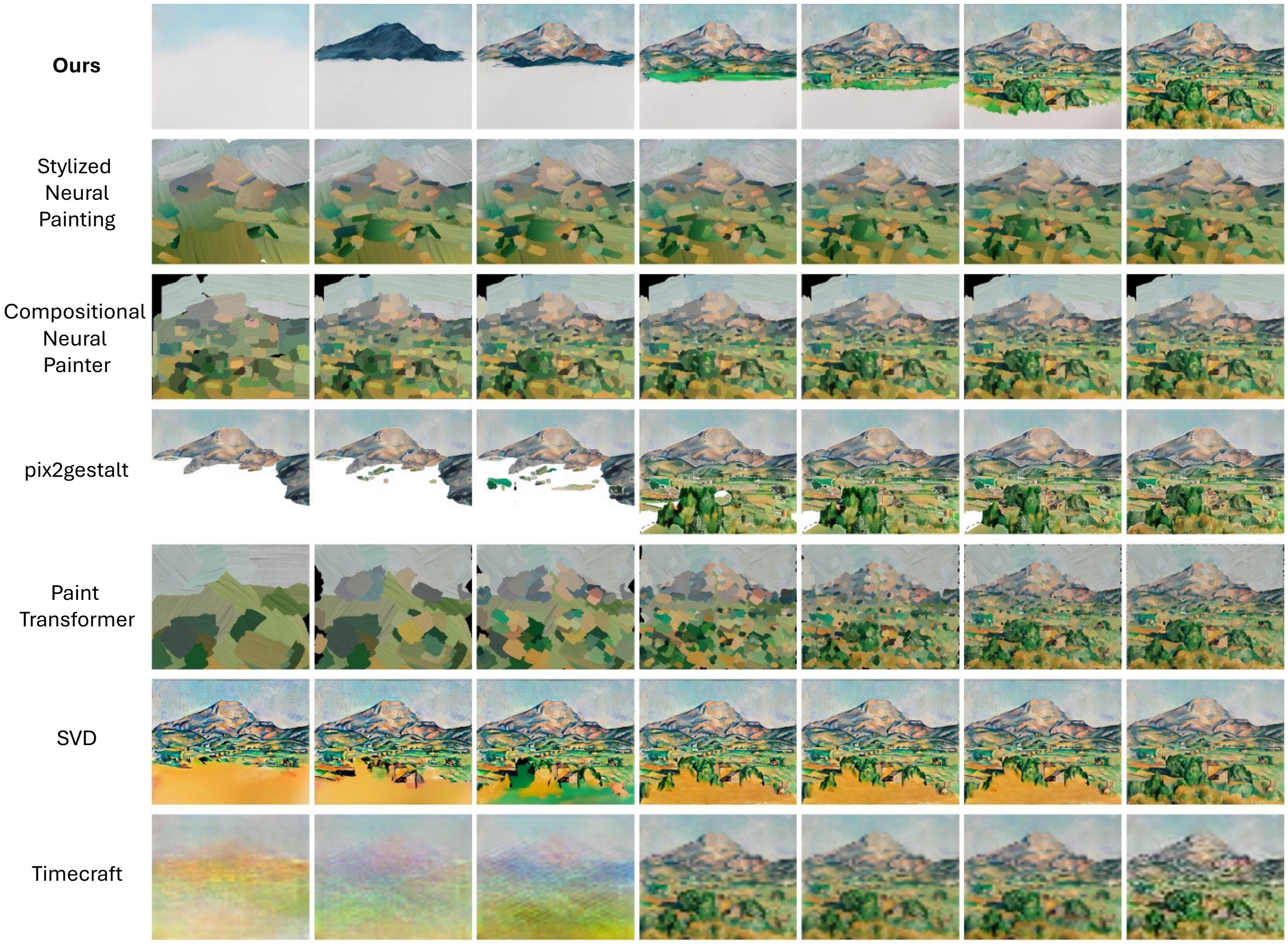


Fig. 12. **Comparison with baselines on in-the-wild images.** Our method significantly outperforms these baselines in generating a more human-like painting video with better visual quality. Images courtesy Barnes Foundation.

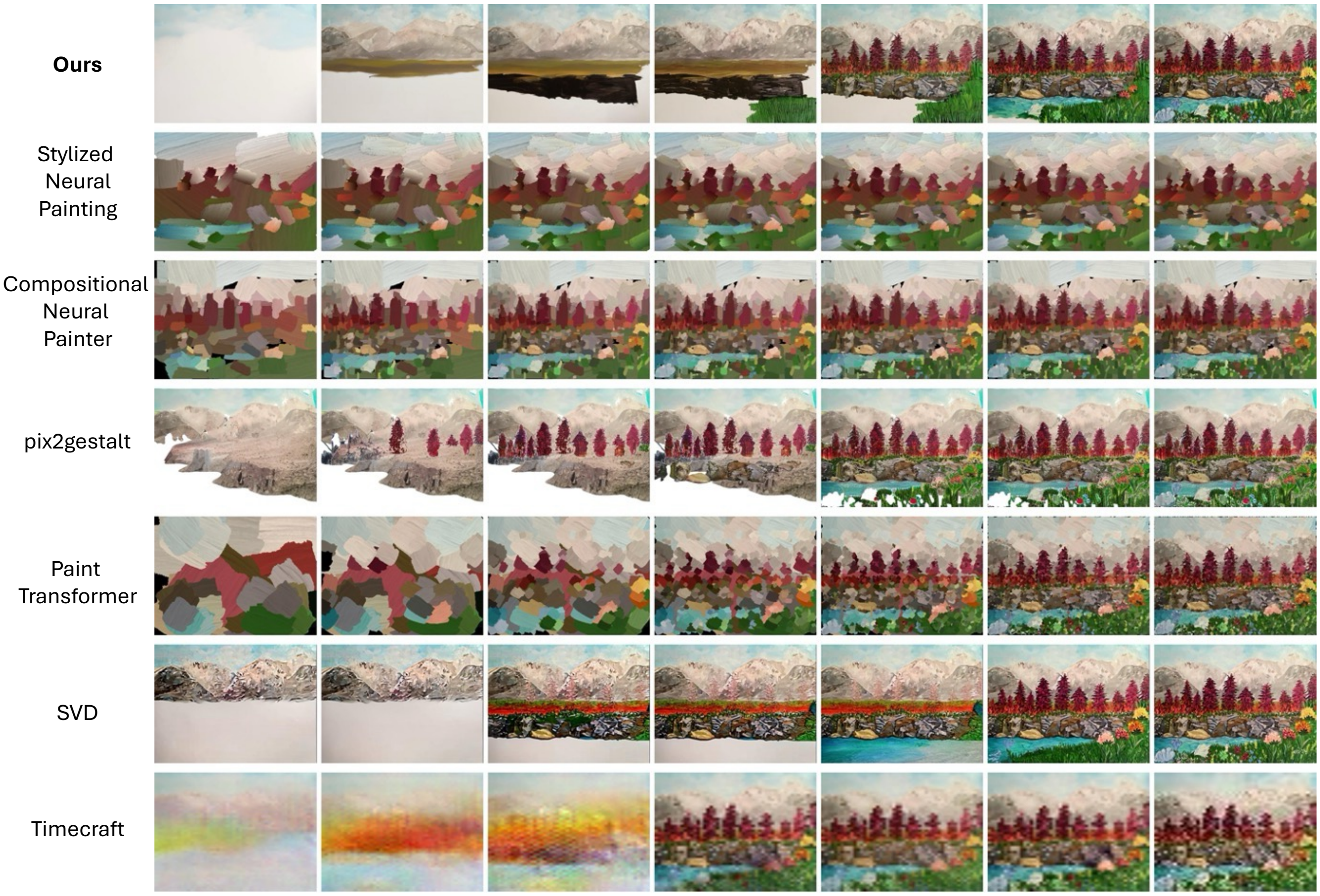


Fig. 13. **Comparison with baselines on in-the-wild images.** Our method significantly outperforms these baselines in generating a more human-like painting video with better visual quality. Image courtesy Catherine Kay Greenup.

REFERENCES

Songwei Ge, Aniruddha Mahapatra, Gaurav Parmar, Jun-Yan Zhu, and Jia-Bin Huang. 2024. On the Content Bias in Fréchet Video Distance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. 2023a. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. *arXiv preprint arXiv:2311.17117* (2023).

Teng Hu, Ran Yi, Haokun Zhu, Liang Liu, Jinlong Peng, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. 2023b. Stroke-based Neural Painting and Stylization with Dynamically Predicted Painting Region. In *Proceedings of the 31st ACM International Conference on Multimedia*. 7470–7480.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems* 35 (2022), 26565–26577.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. 2023. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499* (2023).

Ege Ozguroglu, Ruoshi Liu, Dídac Surś, Dian Chen, Achal Dave, Pavel Tokmakov, and Carl Vondrick. 2024. pix2gestalt: Amodal Segmentation by Synthesizing Wholes. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024).

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695.

SG_161222. 2023. Realistic Vision V5.1.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. In *CVPR*.

Zhengxia Zou, Tianyang Shi, Shuang Qiu, Yi Yuan, and Zhenwei Shi. 2021. Stylized neural painting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15689–15698.