



# Loomis Painter: Reconstructing the Painting Process

Markus Pobitzer<sup>1</sup> Chang Liu<sup>1,†</sup> Chenyi Zhuang<sup>1</sup> Teng Long<sup>1</sup> Bin Ren<sup>1,2</sup> Nicu Sebe<sup>1</sup>

<sup>1</sup>University of Trento      <sup>2</sup>University of Pisa

markus.pobitzer@unitn.it, <sup>†</sup>Corresponding author

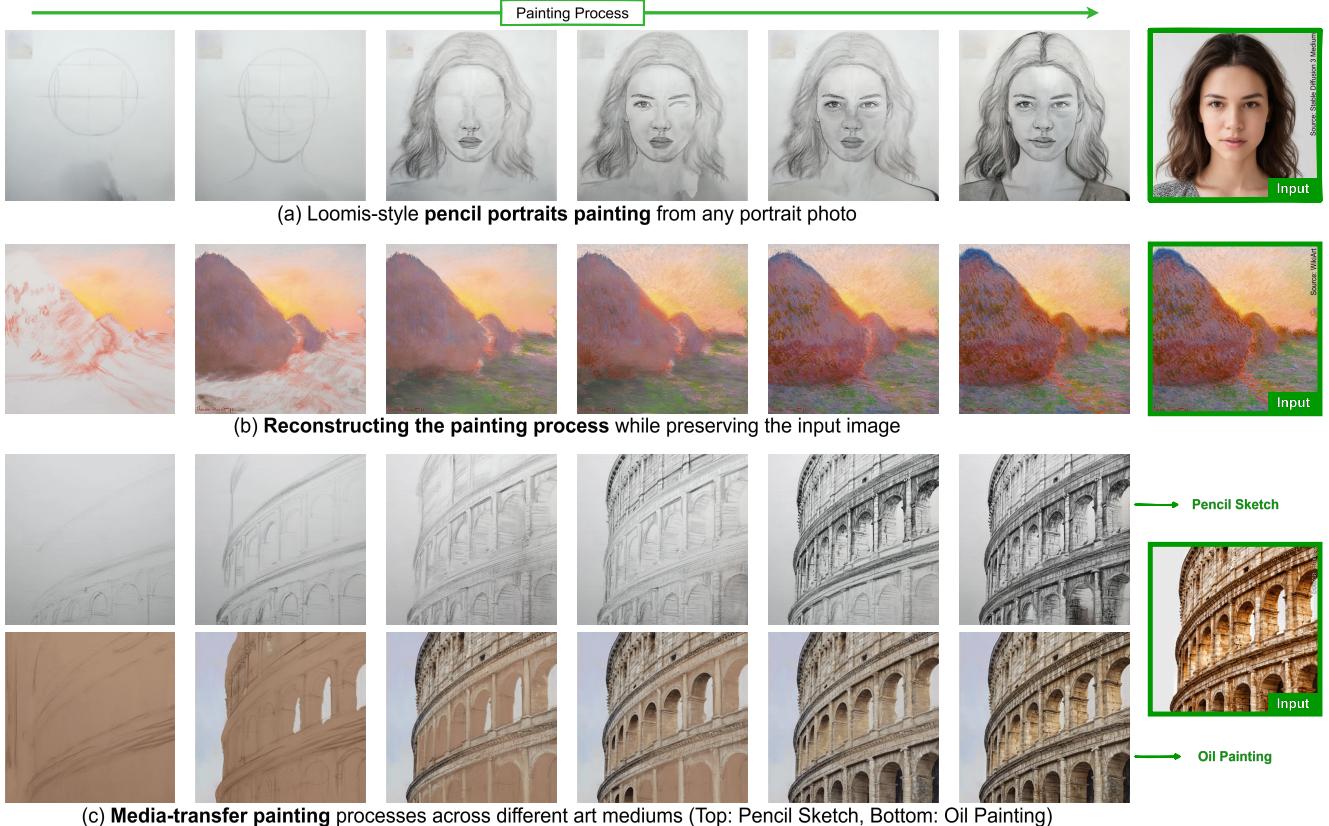


Figure 1. Loomis Painter: Our method reconstructs the painting process of any input image, either faithfully, as shown in (b), or in different art media, as in (a) and (c). The title of our work is inspired by the Loomis portrait method, which we also enable. Images with green borders are input reference images; all others are generated by our method.

## Abstract

*Step-by-step painting tutorials are vital for learning artistic techniques, but existing video resources (e.g., YouTube) lack interactivity and personalization. While recent generative models have advanced artistic image synthesis, they struggle to generalize across media and often show temporal or structural inconsistencies, hindering faithful reproduction of human creative workflows. To address this, we propose a unified framework for multi-media painting process generation with a semantics-driven style control mech-*

*anism that embeds multiple media into a diffusion model’s conditional space and uses cross-medium style augmentation. This enables consistent texture evolution and process transfer across styles. A reverse-painting training strategy further ensures smooth, human-aligned generation. We also build a large-scale dataset of real painting processes and evaluate cross-media consistency, temporal coherence, and final-image fidelity, achieving strong results on LPIPS, DINO, and CLIP metrics. Finally, our Perceptual Distance Profile (PDP) curve quantitatively models the creative sequence, i.e., composition, color blocking, and detail refine-*

ment, mirroring human artistic progression.

## 1. Introduction

Sketching and painting are creative processes that are hard to master. Many tutorials exist as step-by-step guides in the form of books and videos. However, books cannot show the complete dynamic process, and existing instructional videos are inherently passive, lacking interactivity and personalized guidance. While recent generative models have demonstrated impressive capabilities in artistic image synthesis [2, 25, 29], they remain limited in process-level modeling: generated painting sequences often exhibit temporal discontinuities, structural jumps, and poor generalization across artistic media, making it difficult to faithfully reconstruct the coherent, sequential nature of human painting processes.

For artists, it is not only important to replicate a scene but also to bring what they perceive to the canvas with their chosen tools. This includes media-specific steps such as layering color with oil [26] and faithfully reconstructing proportions. A prominent example is the Loomis Method [17], developed by Andrew Loomis, which demonstrates a structural approach to drawing a head with correct proportions. This method can be applied to an existing reference photo, bridging the gap between a static representation and an artistic process.

To address these challenges, we propose a unified framework for multi-media painting process generation, capable of modeling the evolution of artistic content across diverse traditional painting media. We introduce a control mechanism that embeds multiple artistic media into the conditional space of a diffusion model, enabling the model to capture medium-dependent textural evolution and procedural patterns within a shared latent space. Combined with cross-media style augmentation, this framework supports controllable transfer of painting procedures across artistic media while maintaining consistency and coherence in the generated sequences. Another contribution of our work is a reverse-painting training strategy, which learns to regress from the completed artwork back to a blank canvas. This formulation allows the model to learn how to uncover the painting from one generated frame to the next, while the first frame is grounded by the input image. By reversing the direction of the training trajectory and building on top of a video generation model, structural discontinuities and temporal jumps commonly observed in conventional forward-sequence prediction models do not show up. To support high-fidelity learning of real artistic workflows, we construct a large-scale dataset of real drawing and painting processes spanning multiple artistic media. Importantly, we introduce an automatic occlusion-removal procedure that eliminates hand occlusions and other visual clutter, allow-

ing the model to learn accurate stroke-level transformations throughout the creative process.

We conduct extensive experiments evaluating cross-media consistency, temporal coherence, and final-image fidelity. Our method achieves strong performance on LPIPS [37], DINO [18], and CLIP [22] metrics. Beyond conventional evaluations, we propose a new measure, Perceptual Distance Profile (PDP), that quantitatively models structural progression over time. PDP provides a principled way to characterize the trajectory of perceptual changes across frames and reveals that our model closely follows the human creative pattern of composition, color blocking, and detail refinement. To sum up, our contributions are:

- We introduce Loomis Painter, a painting video generation model empowered by our dataset. We enhance the model to enable inference-time translation of diverse scenes into a variety of artistic media, thereby bridging the gap between input and artistic expression.
- We show that the proposed reverse-painting strategy is crucial to accurately reconstructing the input painting.
- We collect and curate a high-quality painting process video dataset that addresses the occlusion problem and covers diverse artistic media and styles beyond existing datasets.
- We extensively evaluate our method with state-of-the-art methods, introducing a novel video-level evaluation for the realism and quality of the generated painting process.

## 2. Related Work

**Neural Painting.** Neural painting aims to reconstruct visual imagery through a sequence of brushstrokes inferred by a neural network. Early efforts, such as Paint Transformer [15], employed feed-forward architectures based on Transformers [4, 30] to progressively generate stroke parameters. Subsequent works such as [21] also include human interaction in the generation process, similar to [3] which lets the user compose an artwork with the help of the neural painter and leverages diversity with a Diffusion Process [8, 24]. Despite their progress, these approaches largely model painting as a parametric rendering problem rather than an authentic artistic process. Their synthesized stroke sequences often diverge from how human artists construct compositions in real tutorials. In contrast, our method directly learns from real-world painting sessions, capturing genuine artistic decision-making and temporal workflows. By operating in pixel space through a video diffusion framework, it overcomes the limitations of parameterized strokes and produces temporally coherent, photorealistic painting progressions akin to those created by skilled artists.

**Pixel-Based Generation.** Pixel-based methods synthesize painting sequences directly at the pixel level, conditioned on a reference image. Early convolutional methods [38] attempted to reconstruct painting workflows, while a more

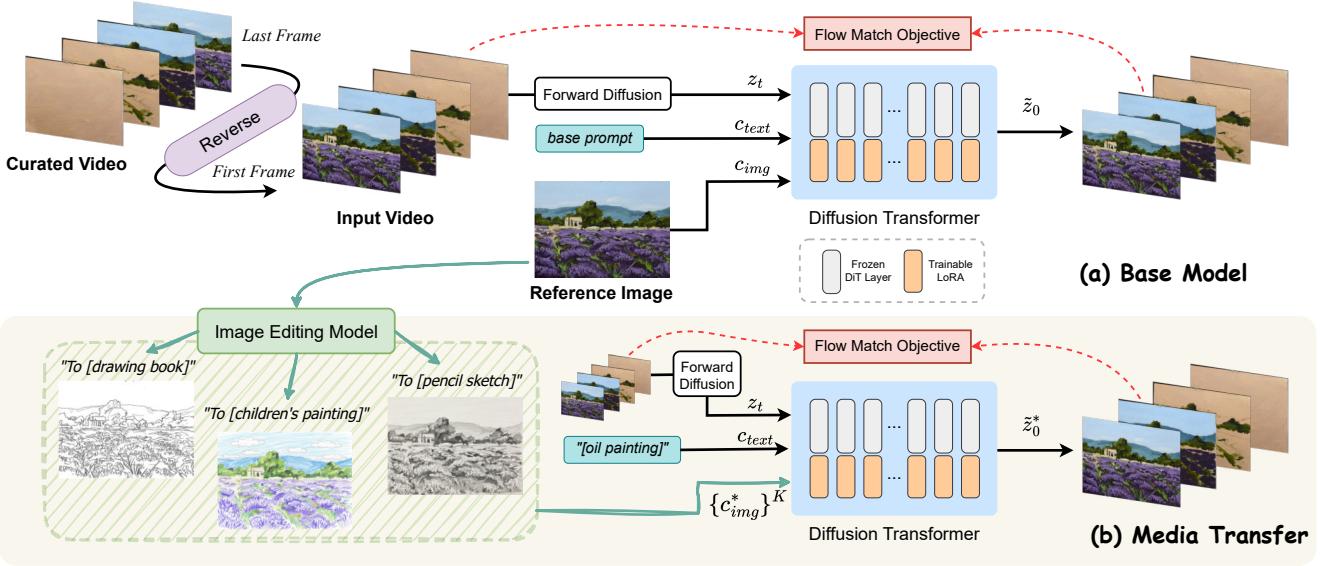


Figure 2. Overview of our painting process generation method. The curated video is first reversed to better align with the underlying video generation model. We LoRA-tune WAN 2.1 [31], a video generation model conditioned on an input image and a prompt. In our case, the input image corresponds to a painting, and the model learns to reconstruct the steps to paint it, starting from a finished painting to a blank canvas. In (b), the media transfer model is shown, which enables the video generation model to render any input image as an acrylic, oil, or pencil painting based on the text input. To achieve this, we generate variations of the reference image using image editing models and train the video generation model to reconstruct the original painting process.

recent work Inverse Painting [2] advanced this painting workflow reconstruction with an autoregressive three-stage pipeline that compares intermediate frames with the reference, masks the next operation area, and updates pixels through diffusion. More recent diffusion-based approaches further push this paradigm: ProcessPainter [25] leverages an image diffusion model to generate the painting process; they mainly use synthetic data for training and only in the last step use a small number of human paintings. PaintsUndo [29] focuses on recreating the painting process for anime-style paintings, leveraging Stable Diffusion [23] as its backbone, and PaintsAlter [36] extends the idea to video diffusion for more continuous progression. Our method builds on a pretrained video diffusion generator to model painting as a *temporally coherent process*. Prior pixel-based approaches, such as Inverse Painting, ProcessPainter, and PaintsUndo, rely on synthetic or narrowly scoped datasets. In contrast, our model can smoothly interpolate from a blank canvas to a completed painting and generalize across diverse artistic media, offering fine-grained control over realistic painting workflows from start to completion.

**Diffusion Models for image and video generation.** Diffusion Models such as Stable Diffusion [23] and FLUX [11] have shown great image generation capabilities. A key innovation in these models is the use of a latent space representation, which significantly reduces the computational cost of the diffusion process. For the image generation

task a text prompt gets leveraged as guidance. In image-to-image tasks, such as inpainting [23], an input image serves as an additional conditioning signal. These concepts have been extended to the video domain [1, 33]. Newer models use the Flow Matching [13] principle to denoise the latents. Compared to the image generation models, the latent vector contains several frames, and there is a temporal compression that enables coherent multi-frame generation. Building upon these foundations, pretrained diffusion models can be fine-tuned using various lightweight techniques to adapt them to new concepts or personalize outputs, e.g., LoRA [9], ControlNet [35], and IPAdapter [34], all of which are more computationally efficient than training a model from scratch.

### 3. Methods

An overview of our method can be seen in Fig. 2.

Our method generates temporally coherent painting sequences that mimic realistic artistic workflows and enable cross-medium transfer. It combines two process-oriented components: cross-media conditioning, which infuses medium-aware semantics to guide strokes and textures, and reverse-painting learning, which aligns temporal supervision with human intuition for a progressive buildup from structure to detail. Given an input prompt and a reference image, the model constructs a semantic condition, evolves a latent temporal representation under the control of

our modules, and finally decodes a complete painting process sequence.

### 3.1. Video Diffusion

Our method is based on a pretrained Video Diffusion model [31] consisting of a video-VAE to encode a given video  $V \in \mathbb{R}^{T \times H \times W \times 3}$  from pixel space into latent space  $x \in \mathbb{R}^{T/C_T \times H/C_H \times W/C_W \times D}$ , where  $C_T$ ,  $C_H$  and  $C_W$  correspond to a temporal, height, and width compression ratio respectively. Typically [7, 20, 31, 33], the temporal compression  $C_T$  is set to 4 or 8, while the spatial ratios  $C_H, C_W$  are 8 or higher, resulting in a highly compact latent representation. This latent is used to train a Diffusion Transformer (DiT) [19] with the Flow Matching objective [6, 13]. Given a video latent  $x_1$ , random noise  $x_0 \sim \mathcal{N}(0, \mathbf{I})$  and timestep  $t \in [0, 1]$ , a linear interpolation  $x_t = tx_1 + (1-t)x_0$  can be defined. The velocity related to  $t$  is  $v_t = x_1 - x_0$ . If we model the output of the DiT as this velocity vector, it enables us to formulate a loss function as the mean squared error (MSE) between the model output and  $v_t$ ,

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, \mathbf{c}_{\text{text}}, t} \|u(\mathbf{x}_t, \mathbf{c}_{\text{text}}, t; \theta) - \mathbf{v}_t\|_2^2, \quad (1)$$

where  $\mathbf{c}_{\text{text}}$  is the text embedding sequence,  $\theta$  denotes the model parameters, and  $u(\mathbf{x}_t, \mathbf{c}_{\text{text}}, t; \theta)$  is the predicted velocity.

Building on text-to-video models, image-to-video (I2V) approaches [12, 20, 31] extend an input image  $I \in \mathbb{R}^{H \times W \times 3}$  into a sparse video tensor  $V_I \in \mathbb{R}^{T \times H \times W \times 3}$  by placing  $I$  in the first frame and padding the remaining  $T - 1$  frames with zeros. This tensor is encoded by a VAE into a condition latent, which is concatenated with the noise latent  $x_t$  along the channel axis to guide synthesis. The combined latent is then processed by the DiT.

### 3.2. Art Media Aware Painting Process

A central challenge in artistic process generation is not only producing distinct visual styles but also reproducing the procedural evolution characteristic of different artistic media. To enable medium-aware process control, we introduce a semantic conditioning mechanism that integrates textual medium attributes into the temporal generative process and aligns them with consistent structural cues across media.

#### 3.2.1. Medium-Aware Semantic Embedding

Given a textual description of the artistic medium  $m$  (e.g., “oil painting”, “pencil sketch”) and a scene description  $s$ , we construct a combined semantic prompt  $p = [m; s]$ . A pretrained text encoder  $E_{\text{text}}(\cdot)$  transforms it into a semantic embedding

$$\mathbf{c}_{\text{text}} = E_{\text{text}}(p) \in \mathbb{R}^d, \quad (2)$$

which serves as a conditioning vector for the generative model.

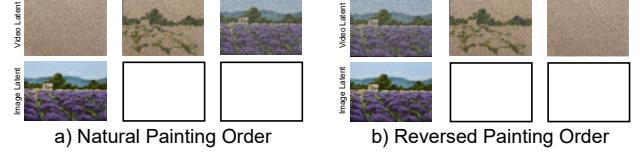


Figure 3. We visualize the noisy video latent and the image latent prior to channel-wise concatenation. Empty boxes indicate the padded frames with zeros. Notably, the natural painting order (a) exhibits poor temporal alignment with the video latents.

During diffusion-based temporal generation,  $\mathbf{c}_{\text{text}}$  is injected through cross-attention:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \quad (3)$$

where the keys and values are augmented as

$$\mathbf{K} = W_K[\mathbf{h}_t; \mathbf{c}_{\text{text}}], \quad \mathbf{V} = W_V[\mathbf{h}_t; \mathbf{c}_{\text{text}}], \quad (4)$$

allowing medium semantics to directly influence the temporal evolution of latent feature  $\mathbf{h}_t$  at each timestep  $t$ .

This embedding drives both stylistic and procedural characteristics: for instance, the model learns color layering behavior in oil painting or progressive hatching patterns in pencil sketching, enabling medium-appropriate workflow synthesis.

#### 3.2.2. Cross-Media Structural Alignment

To enable transferring any input image to a corresponding art medium, we propose a cross-media training strategy. Given an image  $I$ , we apply a style transformation to obtain  $I^*$ , preserving objects and semantics while removing the identity of the original art medium. The standard image-to-video (I2V) loss for an input  $I$  is defined as:

$$\mathcal{L}_{I2V} = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, I, \mathbf{c}_{\text{text}}, t} \|u(\mathbf{x}_t, I, \mathbf{c}_{\text{text}}, t; \theta) - \mathbf{v}_t\|_2^2, \quad (5)$$

For cross-media training, we keep the same target video latent  $x_1$  but replace  $I$  with its transformed counterpart  $I^*$ :

$$\mathcal{L}_{I2V} = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, I^*, \mathbf{c}_{\text{text}}, t} \|u(\mathbf{x}_t, I^*, \mathbf{c}_{\text{text}}, t; \theta) - \mathbf{v}_t\|_2^2, \quad (6)$$

This strategy exposes the model to consistent shapes, contours, and spatial relationships across different styles, enabling it to learn how these elements map to the target artistic medium. Each object is progressively rendered over time, simulating a natural painting process. Consequently, the model learns to translate an arbitrary input image into a procedural painting trajectory defined by the specified medium.

### 3.3. Reverse-Painting Learning Strategy

Following the natural painting order, from blank canvas to finished artwork, introduces two key challenges.

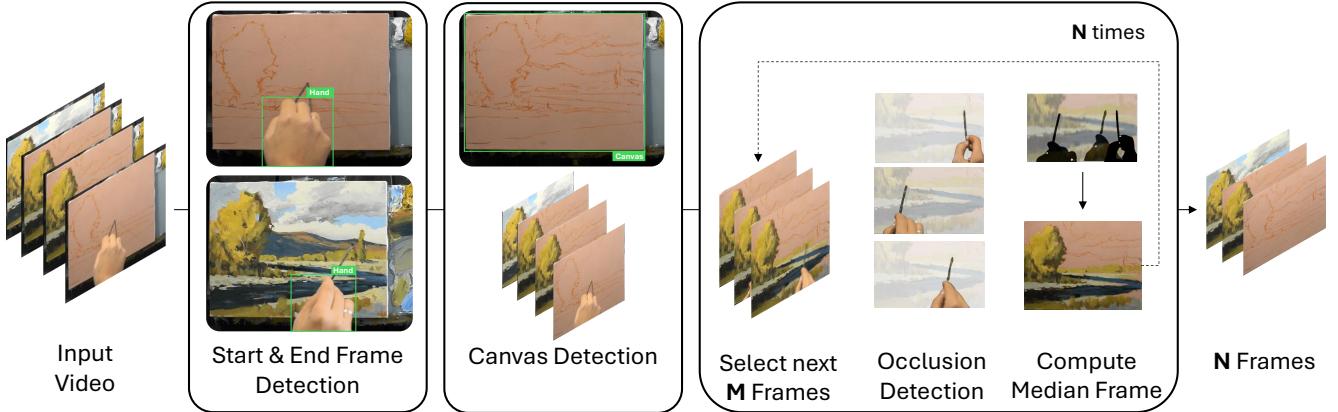


Figure 4. Dataset Curation Pipeline Overview. Our framework extracts painting workflows from raw tutorial videos. First, start and end frames are detected, and the painting canvas is localized. The video is then partitioned into  $N$  segments, from which  $M$  frames are sampled per segment. Subsequently, occlusions (e.g., hands, brushes) are detected in the sampled frames, and a masked median is computed over the sampled frames, using the preceding median frame as a reference to remove transient obstructions. Finally, logos and text overlays are detected and removed (not shown in the figure), producing  $N$  occlusion-free frames. Image courtesy of Samir Godinjak, from the Painting with Samir YouTube channel.

First, existing I2V models [12, 20, 31] are trained to reconstruct the input image in the initial frame, which in our case corresponds to the completed painting. Generating a blank canvas first would require substantial retraining to override this default behavior.

Second, the input image latent is temporally misaligned with the generation process. As discussed in Sec. 3.1, video diffusion models concatenate the padded image latent with the noise latent along the channel axis. However, the image latent is typically placed at the first temporal position, creating a mismatch between conditioning and the intended progressive painting trajectory, see Fig. 3.

We propose a reverse-painting learning strategy that reorganizes temporal supervision to achieve smoother procedural modeling. Instead of predicting the next stroke forward in time, the model learns to gradually reveal the painting in reverse order. For a video diffusion model, this formulation emphasizes consistency with the previous temporal frame, allowing the network to focus on reconstructing what has already been partially revealed rather than anticipating future strokes.

**Temporal reversal.** Given an original painting video  $V_{og} = \{f_1, f_2, \dots, f_T\}$  that depicts the progression from an empty canvas to a completed artwork, we construct its reversed sequence:

$$V_{rev} = \{f_T, f_{T-1}, \dots, f_1\}.$$

This reversal naturally introduces a monotonic ‘‘detail removal’’ process: high-frequency textures gradually fade, color regions simplify into coarse structural blocks, and the underlying composition becomes increasingly dominant.

Table 1. Overview of the curated video dataset by art medium.

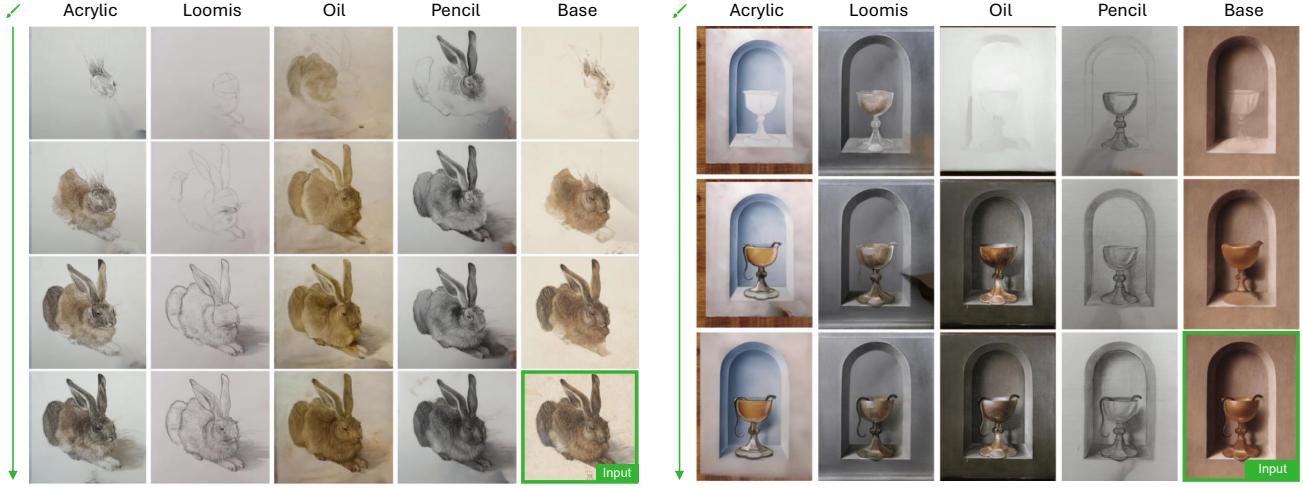
	Acrylic	Oil	Pencil	Loomis	Total
# of Videos	81	151	298	207	<b>737</b>
Avg. Duration [min]	$\approx 40$	$\approx 30$	$\approx 12$	$\approx 20$	—

## 4. Dataset Curation Pipeline

An overview of our pipeline is shown in Fig. 4. Our process begins with temporal trimming, where we detect the first and last appearance of a ‘‘hand’’ using GroundingDINO [16] to isolate the core painting process. Next, for canvas localization, we first attempt to find the ‘‘canvas’’ using GroundingDINO; for split-screen tutorials (e.g., Loomis), we instead compute the maximum horizontal intensity gradient to separate the reference photo from the canvas. We then partition the video into 10-second segments, sampling 30 frames (3fps) from each. Occlusions (e.g., hands, brushes) are segmented using InSPyReNet [10] or BiRefNet [39]. A clean frame for each segment is generated by computing a masked median of its samples; this calculation iteratively incorporates prior frames to fill persistent occlusions. Finally, in post-processing, we detect logos and text with GroundingDINO [16] and inpaint them using LaMa [28].

**Data Collecting.** We curated a dataset from painting tutorial videos on YouTube, prioritizing static camera angles and minimal canvas movement. All videos were processed with the pipeline described in Sec. 4 and manually reviewed to filter out poor-quality results. A detailed breakdown of our final dataset is provided in Tab. 1.

**Fine-Tuning Datasets.** We merge the Acrylic, Loomis Portrait, Oil, and Pencil subsets into a unified dataset for generalizable fine-tuning. Each entry includes a reference



(a) The input image is called Hare by Albrecht Durer image courtesy of WikiArt. In the case of art media transfer, we used the following prompt with the appropriate art medium inserted: "<art\_medium>Step by step painting process. Create an image of a brown rabbit with long ears and a fluffy coat, sitting on a white surface with a shadow cast beneath it."

(b) Input image is St. John and Veronica Diptych (reverse) by Hans Memling image courtesy Wikiart. In the case of art media transfer, we used the following prompt with the appropriate art medium inserted: "<art\_medium>Step by step painting process. Create an image of a golden goblet with a snake coiled around its handle, set against a gray stone archway."

Figure 5. Comparison using the same input image (bottom right). Columns 1–4 show samples from the art media transfer model; column 5 shows the base model output. As the base model’s final frame closely matches the input, only the input image is shown. For the base model, we employed the standard prompt. The last row shows the final frame of each method.

frame (the finished artwork) and its progressive painting states. We will release the code and configuration files required to reproduce our dataset, but not the data itself due to licensing constraints.

## 5. Experiments

**Implementation Details.** We extend the Wan 2.1 14B 480p I2V model [31] as our base video generation model. Wan does not temporally compress the first generated frame, enabling more accurate reconstruction. This design aligns naturally with our reverse-painting strategy, as the first frame corresponds to the input image. During fine-tuning, all images are resized and padded to a resolution of 480×832 pixels. The model can be fine-tuned with LoRA adapters on the dataset in 24h on 4 Nvidia H100 GPUs, learning rate of  $1e - 4$ . This corresponds to 14 training epochs, which we found to yield the best performance in our experiments. We use the dateset described in Sec. 4 to train the LoRA that was used for the evaluations. To enable painting media transfer we train a separate LoRA, starting from our base model trained for 7 epochs and train it for an other 7 epochs on the art media transfer dataset. In the Appendix Sec. A.4 more details can be found how the art media transfer dataset was constructed. All datasets used in fine-tuning follow a 90% train split.

### 5.1. Baselines

We compare our approach against three representative methods: Inverse Painting [2], which autoregressively re-

Table 2. Evaluation based on similiarity metrics

Method	FID ↓	LPIPS ↓	Clip ↑	Dinov2 ↑
Inverse Painting	326.15	0.61	0.66	0.21
ProcessPainter	282.90	0.53	0.76	0.50
PaintsUndo	236.52	0.55	0.77	0.56
Abl. 7 epochs	172.62	0.42	0.84	0.72
Ours 7 epochs	164.29	0.39	0.85	0.73
Ours	<b>151.04</b>	<b>0.38</b>	<b>0.86</b>	<b>0.76</b>

constructs the painting process by determining which area in the painting should be filled next and inpainting the selected area with a diffusion model.

ProcessPainter [25], which learns to reconstruct the painting process using only a few real painting examples.

PaintsUndo [29], designed for reconstructing the painting process with fine-grained control over the painting progression.

### 5.2. Quantitative Comparison

We evaluate our method against state-of-the-art baselines using LPIPS, CLIP and Dinov2 (to assess perceptual similarity to human workflows) and FID [5] (to measure distributional alignment with ground-truth painting sequences). For each method, generated frames are compared to the closest corresponding ground-truth frame in terms of painting progression. Metrics are first averaged across frames

Table 3. Perceptual Distance Profile (PDP) Evaluation.

Method	PDP [LPIPS] ↓			PDP [Clip] ↓			PDP [Dinov2] ↓		
	pdp	pdp norm	distance	pdp	pdp norm	distance	pdp	pdp norm	distance
Inverse Painting [2]	0.320	1.075	0.653	0.128	1.137	0.190	0.309	1.170	0.412
ProcessPainter [25]	0.174	0.176	0.262	0.067	0.203	0.061	0.131	0.243	0.079
PaintsUndo [29]	0.162	0.176	0.218	0.055	0.267	0.053	0.117	0.315	0.062
Ours	<b>0.098</b>	<b>0.160</b>	<b>0.122</b>	<b>0.031</b>	<b>0.199</b>	<b>0.033</b>	<b>0.072</b>	<b>0.184</b>	<b>0.027</b>

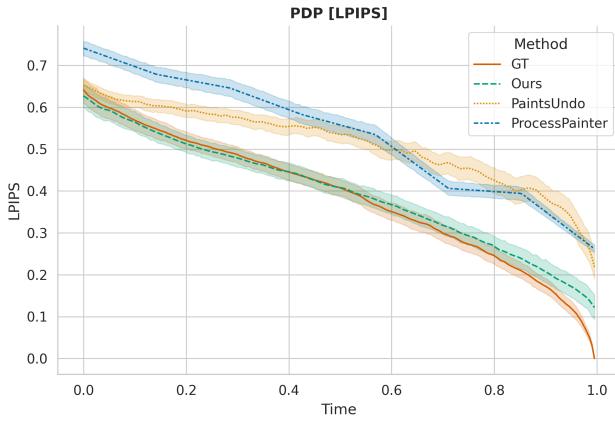


Figure 6. Average of the PDP curve of our test set comparing different methods. LPIPS was used as perceptual metric. The gap at  $Time = 1$  of a method and the ground truth stems from the fact that the methods can not loss less reconstruct the input image.

within each test video, and then averaged across all test videos. This approach ensures that poor performance on individual videos is appropriately reflected in the overall evaluation. As shown in Tab. 2, our method achieves the best performance across all metrics.

### 5.2.1. Perceptual Distance Profile

To evaluate the temporal consistency and plausibility of the painting process, we introduce the Perceptual Distance Profile (PDP), a novel metric designed to compare the sequence of a generated video against its ground truth counterpart. Standard frame-based metrics fail to capture the process of creation, which is a key aspect of our task. The PDP addresses this by measuring the perceptual distance of every frame to the final, completed painting. Our method, detailed in Algorithm 1 (see Appendix), compares this distance profile of the generated video to the ground truth. The profiles are interpolated onto a common, normalized time axis and the L2 distance between them is taken as the final pdp score. The PDP score itself is computed individually for each video pair, providing a precise per-sample evaluation. The metric is modular, allowing any underlying perceptual distance function (*e.g.*, LPIPS [37], DINO [18], CLIP [22]) to be used and can compare videos of different frame lengths. A plot of the perceptual distance to the final

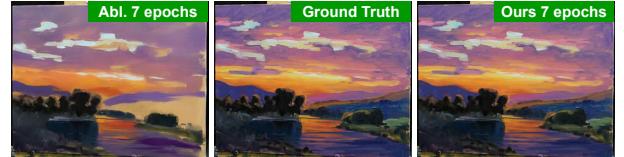


Figure 7. Comparison of final frame. On the left, we see that the Ablation is not able to reconstruct the input image fully; several details are still missing, *i.e.*, the bottom right part is not complete. Image courtesy Samir Godinjak, from the Painting with Samir YouTube channel.

image is shown in Fig. 6.

We report PDP scores in Tab. 3. The scores are computed on generated, ground truth video pairs and averaged over all videos. We indicate with pdp the general score, with pdp normd, the score where the start and end points of the profile are normalized, this helps to focus only on the painting process. Finally, distance indicates the averaged perceptual distance between the last generated frame and the input frame. The distance should be as close as possible to 0, indicating the capabilities of correctly reconstructing the input image. Our method shows overall strong performance.

### 5.3. Ablation

To demonstrate that the un-painting (reverse) frame order during LoRA tuning yields better results we fine-tuned the video generation model on both frame ordering. Quantitative results can be seen in Tab. 2 where Abl. 7 epochs indicates the painting order as seen in painting tutorial videos, from blank canvas to finished painting. Ours 7 epochs indicates the reversed frame ordering that we used in our method. Both models were trained for 7 epochs. Visually, the difference in generation quality can also be seen in Fig. 7 where the Ablation is not able to fully reconstruct the input image.

### 5.4. Qualitative Results

Our base model reliably reconstructs a wide range of input images, as illustrated in the teaser Fig. 1.

The teaser also showcases the Loomis method, which generates pencil-style renderings of portrait photographs. Although trained exclusively on human faces, the model

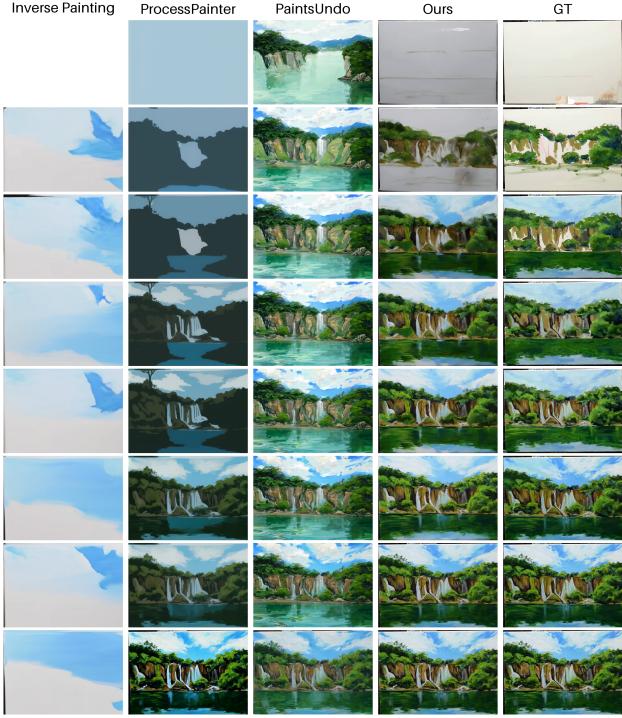


Figure 8. Visualization of ours with other methods

demonstrates strong generalization capabilities, extending to animal heads as well. This is evident in Fig. 5 (a) where the rabbit head is segmented into regions to facilitate structured drawing.

Our method supports rendering the same input image in various art media as shown in Fig. 5. While the base model aims for faithful reconstruction, the art media transfer model has more artistic freedom, it keeps true to the object but changes color and the painting process to follow the art media specified in the prompt. For example, the hare is rendered in multiple styles, with notable differences between pencil-based media such as Loomis and other artistic formats.

We further provides visualization comparison with other methods in Fig. 8. For a good overview of the capabilities of the method please consider watching the videos on our website.

### 5.5. Qualitative Comparison

Comparison of the methods can be seen in Fig. 8. For all the other methods we evenly sampled the shown frames. Input frame for the methods is the last shown ground truth (GT) frame.

Inverse Painting often was not able to converge to the input frame and got stuck by painting the sky or background as can be seen in this example.

ProcessPainter shows different layers of the painting, however the steps appear synthetic and not like real stroke

orders. The number of generated frames is also limited to eight frames making it a coarse painting process.

PaintsUndo demonstrates more coherent painting sequences. Sometimes, it tends to progress rapidly in the early stages, and there are some inconsistencies between frames.

Our method closely mirrors the ground truth. It begins with a structural sketch, gradually layers in color, and incrementally adds detail to reach the final painting. This progression aligns well with human painting workflows and demonstrates the model’s ability to synthesize temporally coherent and stylistically faithful sequences.

## 6. Conclusion

In this work, we have shown that a video diffusion model can successfully recreate the painting process of a given image. Our main focus was on traditional painting media, for which we developed a data collection pipeline and curated a diverse dataset.

The trained model demonstrates a deep understanding of the painting process by sketching outlines, deploying hatching techniques in pencil sketches, using layering techniques, and showing a natural understanding of objects and their depth in the scene, as well as the interactions between shadow and light, among other aspects. To better evaluate the painting process, we introduced the Perceptual Distance Profile (PDP) metric.

## 7. Limitation & Future Work

**Limitations.** Our occlusion detector in the data collection pipeline cannot detect hand shadows, leading to dark artifacts in the training data. These artifacts are mainly visible in pencil painting generations, typically in the bottom-right region. Our base model struggles to paint portraits since it has never seen them during training. An example can be seen in the Appendix Fig. B where the model tried to move the head of the men during the painting process. The art media transfer model does not exhibit this issue because Loomis pencil paintings were transformed into portrait photos. In certain cases, the art media model fails when generating combinations of content and art media that were not seen during training. Examples include applying the Loomis method to non-portrait drawings (Fig. 5) and using the acrylic method for human portraits, as it was only trained on landscapes (Fig. B, Fig. D).

**Future Work.** To fully support human artists in their painting journey, simply showing a step-by-step sequence is not enough. Understanding the painting process also requires indicating which colors were selected, how they were mixed, which tools (pencils or brushes) were used, and how to apply them on the canvas.

## References

- [1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 3
- [2] Bowei Chen, Yifan Wang, Brian Curless, Ira Kemelmacher-Shlizerman, and Steven M Seitz. Inverse painting: Reconstructing the painting process. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 2, 3, 6, 7
- [3] Nicola Dall’Asen, Willi Menapace, Elia Peruzzo, Enver Sangineto, Yiming Wang, and Elisa Ricci. Collaborative neural painting. *CVIU*, page 104298, 2025. 2
- [4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 2
- [5] D.C Dowson and B.V Landau. The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. 6
- [6] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 4
- [7] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024. 4
- [8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2
- [9] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3
- [10] Taehun Kim, Kunhee Kim, Joonyeong Lee, Dongmin Cha, Jiho Lee, and Daijin Kim. Revisiting image pyramid structure for high resolution salient object detection. In *ACCV*, pages 108–124, 2022. 5, 1
- [11] Black Forest Labs. Flux. [urlhttps://github.com/black-forest-labs/flux](https://github.com/black-forest-labs/flux), 2024. 3
- [12] Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shanghai Yuan, Luhuan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024. 4, 5
- [13] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 3, 4
- [14] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 2
- [15] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. Paint former: Feed forward neural painting with stroke prediction. In *ICCV*, pages 6598–6607, 2021. 2
- [16] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, pages 38–55. Springer, 2024. 5, 1
- [17] Andrew Loomis. *Drawing the head & hands*. Clube de Autores, 2021. 2, 1
- [18] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 7
- [19] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 4
- [20] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, et al. Open-sora 2.0: Training a commercial-level video generation model in \$200 k. *arXiv preprint arXiv:2503.09642*, 2025. 4, 5
- [21] Elia Peruzzo, Willi Menapace, Vudit Goel, Federica Arrigoni, Hao Tang, Xingqian Xu, Arman Chopikyan, Nikita Orlov, Yuxiao Hu, Humphrey Shi, et al. Interactive neural painting. *Computer Vision and Image Understanding*, 235: 103778, 2023. 2
- [22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2, 7
- [23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 3, 2
- [24] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265. pmlr, 2015. 2
- [25] Yiren Song, Shijie Huang, Chen Yao, Xiaojun Ye, Hai Ci, Jiaming Liu, Yuxuan Zhang, and Mike Zheng Shou. Processpainter: Learn painting process from sequence data. *arXiv preprint arXiv:2406.06062*, 2024. 2, 3, 6, 7
- [26] H. Speed. *Oil Painting Techniques and Materials*. Dover Publications, 1987. 2
- [27] Emilian Stevenson. Epicrealism, 2023. 2
- [28] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *WACV*, pages 2149–2159, 2022. 5, 1
- [29] Paints-Undo Team. Paints-undo github page, 2024. 2, 3, 6, 7
- [30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

- [31] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 3, 4, 5, 6
- [32] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025. 1
- [33] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 3, 4
- [34] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 3
- [35] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 3
- [36] Lvmin Zhang, Chuan Yan, Yuwei Guo, Jinbo Xing, and Maneesh Agrawala. Generating past and future in digital painting processes. *ACM Trans. Graph.*, 44(4), 2025. 3
- [37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2, 7
- [38] Amy Zhao, Guha Balakrishnan, Kathleen M Lewis, Frédéric Durand, John V Guttag, and Adrian V Dalca. Painting many pasts: Synthesizing time lapse videos of paintings. In *CVPR*, pages 8435–8445, 2020. 2
- [39] Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 2024. 5, 1



# Loomis Painter: Reconstructing the Painting Process

## Supplementary Material

### A. Dataset Curation Pipeline

A more in depth discussion about the dataset curation pipeline than in the main paper.

#### A.1. Painting Video Extraction Pipeline

**Temporal Trimming.** Raw tutorial videos often include irrelevant introductory or outro segments. To isolate the painting process, we detect the start frame as the first occurrence of a hand (indicating artist activity) and the end frame as the last hand appearance using GroundingDINO [16]. The video is then trimmed to this interval.

**Canvas Localization.** We localize the canvas using GroundingDINO’s zero-shot object detection, querying for “canvas”. For Loomis portrait tutorials, which typically display a reference photograph on the left and the canvas on the right, we compute horizontal intensity gradients across each frame and split the image at the column with maximal gradient magnitude, isolating the canvas region. This reliably isolates the canvas region, as the reference photograph typically has a dark monotone background, while the canvas is a white sheet, producing a strong gradient.

**Frame Sampling and Occlusion Removal.** The trimmed video is partitioned into  $N = \frac{\text{videoduration(sec)}}{10}$  segments. From each segment, we sample  $M = 30$  frames (3 frames/sec). For each sampled frame we detect occlusions (e.g., hands, brushes) using InSPyReNet [10] or BiRefNet [39], which segment foreground objects via iterative refinement. Afterwards a median frame is computed from the  $M$  samples, masked to exclude occluded pixels. The mask ensures that none of the detected occlusions are part of the final frame. To fill regions persistently occluded in the sample, we iteratively include the median of prior segments in the computation, initializing with a blank white canvas. The process leaves us with  $N$  frames.

**Post-Processing.** Logos and text overlays are detected using GroundingDINO and removed via inpainting with LaMa [28].

**Efficiency.** On an NVIDIA RTX A4000 GPU, our pipeline processes videos with a resolution of 640x360 pixels in near real-time (processing time approximately equals video duration), enabling scalable dataset curation.

### A.2. Data Collecting

We curate a dataset from painting tutorial videos on YouTube, prioritizing videos with static camera angles and minimal canvas movement to simplify temporal alignment. After processing the videos using the pipeline introduced in Sec. A, we manually reviewed the outputs and excluded

those with poor results. In total, we collected 767 videos spanning a variety of art media and artists.

**Acrylic.** This subset includes 81 photorealistic acrylic landscape painting tutorials (avg. 40 min duration), emphasizing techniques like wet-on-wet blending and layering.

**Oil.** We collected 151 oil painting tutorials, including 142 impressionist landscapes (loose brushwork, vibrant palettes) and 9 photorealistic paintings (avg. 30 min duration).

**Pencil.** This subset comprises 270 pencil and 28 colored pencil tutorials (avg. 12 min duration), covering various scenes and motives.

**Loomis Portraits.** We include 207 portrait tutorials following Andrew Loomis’ proportional method [17]. These videos typically display a reference photograph on the left and the canvas on the right. We isolate the canvas via horizontal gradient splitting, as described in Sec. A.

### A.3. Limitations.

Our dataset is biased toward pencil-based painting sequences, with fewer examples of color workflows. The diversity of artists is also limited. A major challenge is that many tutorials include excessive camera movements, zooms, and occlusions, which reduce temporal consistency. The motives are also fixed on landscape drawings and portrait drawings with the loomis method. Despite these constraints, the dataset provides a solid foundation for modeling acrylic, oil, and pencil painting processes.

### A.4. Painting Media Transfer Dataset

Unlike the combined dataset, which uses finished paintings as references, Loomis tutorials require conditioning on real portrait photographs rather than sketches. Similar we want to enable a sort of style/media transfer for the other medias. Given a sketch, generate a oil or acrylic painting and vice versa. To bridge this gap, we synthesize variations fo the reference frame with image editing models. For the pencil sketches and loomis portrait drawings we generate realistic photos and color paintings using ControlNet, ensuring alignment between reference photos and artwork. For acrylic and oil paintings we use Qwen Image Edit [32] to generate pencil sketches, children drawings and drawing book styles.

The tradeoff we noticed is that ControlNet follows exactly the outlines of the reference paintings but loses information such as color due to the underlying control mechanism. On the other side Image Editing models keep this information but the ones we tested slightly change the im-

age composition either due to the models generative capabilities or due to a resolution mismatch. Therefore we used ControlNet for pencil sketches and Qwen Image Edit for the color paintings.

When we generate reference frame variations with ControlNet we use the LineArt processor, it works well for paintings. [27] was used as the image generation model, it is a fine tuned version of Stable Diffusion 1.5 [23].

We generate captions for the original reference frame with LLavaNexT [14], these captions get combined with the art media label. During fine tuning we replace the original frame with the generated frames and the model has to learn the transfer based on the prompt. In Fig. B and Fig. C the prompt is shown in the figure description.

## B. Perceptual Distance Profile

To evaluate the temporal consistency and plausibility of the painting process, we introduce the Perceptual Distance Profile (PDP), a novel metric designed to compare the sequence of a generated video against its ground truth counterpart. Standard frame-based metrics fail to capture the process of creation, which is a key aspect of our task. The PDP addresses this by first establishing a "profile" for each video, which is computed by measuring the perceptual distance of every frame to the final, completed painting. We observe that for our ground truth dataset, the average of these profiles converges to a smooth, characteristic curve, representing a canonical painting process, starting steep, progressing steadily, and finishing with fine details.

The pseudo code to compute the perceptual distance profile is shown in Algorithm 1. Note that for metrics such as DINO or CLIP, we transform the cosine similarity range from  $[-1, 1]$  to a distance range of  $[0, 1]$ , where lower values indicate higher perceptual similarity. The profiles are then interpolated onto a common, normalized time axis. This step makes the metric inherently flexible, as it does not require the two videos to be the same length. The final PDP score is the L2 distance between these two normalized curves, with a lower score indicating that the generated video's painting process is perceptually closer to the ground truth. To focus purely on the process i.e., how quickly perceptual details are added and to handle discrepancies in generated starting frames or imperfect final reconstructions, we also report a normalized score where each profile was normalized to a  $[1, 0]$  range.

## C. Qualitative results

A qualitative comparison with other methods is presented in Fig. A. Further qualitative results on a portrait photo are shown in Fig. B, where our base model struggles to accurately reconstruct the subject's head and exhibits noticeable movement during generation. This artifact is absent

in the art-media model, which benefits from fine-tuning on portrait photos using the Loomis method. In Fig. C, we compare different art media in rendering a castle. Notably, the base model introduces a white background in the final frames to match the input image. Since our training data primarily consists of colored paper or canvas, the model finds it challenging to synthesize a completely white background, a feature common in digital paintings. Conversely, the Loomis art-media variant fails to produce a coherent result in this case. For completeness, the painting process of the Mona Lisa is illustrated in Fig. D.

---

**Algorithm 1** Perceptual Distance Profile (PDP)

---

```

1: Input:  $V_{gt}$ , Ground truth video  $[f_0^{gt}, \dots, f_{T_{gt}-1}^{gt}]$ 
2:       $V_{gen}$ , Generated video  $[f_0^{gen}, \dots, f_{T_{gen}-1}^{gen}]$ 
3:       $\mathcal{D}$ , Perceptual distance function (e.g., LPIPS, DINO)
4:       $N_{points}$ , Number of interpolation points (e.g., 200)
5:      normalize, Boolean flag
6: Output:  $pdp$ , The Perceptual Distance Profile score

7: function NORMALIZEPROFILE( $P$ )
8:    $P_{end} \leftarrow P[\text{last}]$ 
9:    $P_{start} \leftarrow P[\text{first}]$ 
10:   $denominator \leftarrow P_{start} - P_{end}$ 
11:  if  $|denominator| < 10^{-8}$  then
12:     $denominator \leftarrow 1.0$ 
13:  end if
14:   $P_{norm} \leftarrow (P - P_{end}) / denominator$ 
15:  return  $P_{norm}$ 
16: end function

17: function COMPUTEPDP( $V_{gt}, V_{gen}, \mathcal{D}, N_{points}, normalize$ )
18:    $F_{target} \leftarrow V_{gt}[\text{last}]$                                  $\triangleright$  Get the final ground truth frame
19:    $P_{gt} \leftarrow []$                                           $\triangleright$  Compute raw distance profile for ground truth
20:   for each frame  $f$  in  $V_{gt}$  do
21:      $dist \leftarrow \mathcal{D}(f, F_{target})$ 
22:      $P_{gt}.\text{append}(dist)$ 
23:   end for
24:    $P_{gen} \leftarrow []$                                           $\triangleright$  Compute raw distance profile for generation
25:   for each frame  $f$  in  $V_{gen}$  do
26:      $dist \leftarrow \mathcal{D}(f, F_{target})$ 
27:      $P_{gen}.\text{append}(dist)$ 
28:   end for
29:   if normalize then                                          $\triangleright$  Normalize profiles to focus on the process
30:      $P_{gt} \leftarrow \text{NORMALIZEPROFILE}(P_{gt})$ 
31:      $P_{gen} \leftarrow \text{NORMALIZEPROFILE}(P_{gen})$ 
32:   end if                                                  $\triangleright$  Resample both profiles to a common time axis [0, 1]
33:    $t_{gt} \leftarrow \text{Linspace}(0, 1, \text{length}(P_{gt}))$ 
34:    $t_{gen} \leftarrow \text{Linspace}(0, 1, \text{length}(P_{gen}))$ 
35:    $t_{common} \leftarrow \text{Linspace}(0, 1, N_{points})$ 
36:    $C_{gt} \leftarrow \text{LINEARINTERPOLATE}(t_{gt}, P_{gt}, t_{common})$ 
37:    $C_{gen} \leftarrow \text{LINEARINTERPOLATE}(t_{gen}, P_{gen}, t_{common})$            $\triangleright$  Compute L2 distance between the two curves
38:    $diff\_sq \leftarrow (C_{gen} - C_{gt})^2$ 
39:    $integral \leftarrow \text{INTEGRATE}(diff\_sq, \text{using } t_{common})$ 
40:    $pdp \leftarrow \sqrt{integral}$ 
41:   return  $pdp$ 
42: end function

```

---



Figure A. Visual comparison of different methods. First column Inverse Painting, the method did not converge to the reference painting. Second row Process Painter. Third row PaintsUndo, it struggled with the content of the image, most likely due to the fact that it mainly was trained on digital paintings and is not familiar with this rather abstract oil painting. Next column our method and in the last column the ground truth painting process. Image courtesy Samir Godinjak, from the Painting with Samir YouTube channel.

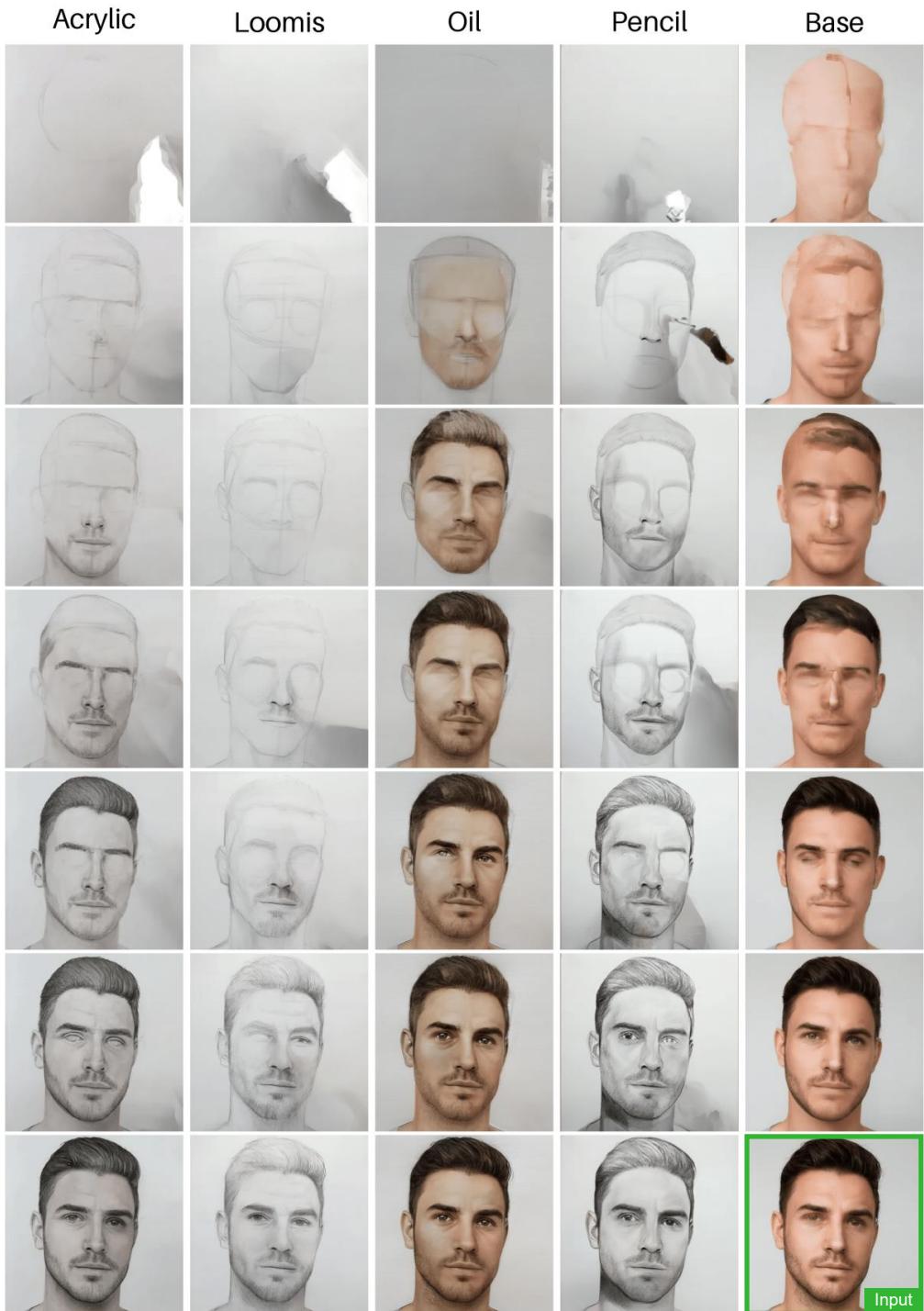


Figure B. Comparison using the same input image (bottom right). Columns 1–4 show linearly sampled outputs from the art media transfer model; column 5 shows the base model output. As the base model’s final frame closely matches the input, only the input image is shown. The input image was generated, image courtesy Stable Diffusion 3-medium. For the base model, we employed the standard prompt. In the case of art media transfer, we used the following prompt with the appropriate art medium inserted: "<art.media>Step by step painting process. The image features a man with short dark hair and a beard, looking directly at the camera with a neutral expression. He is wearing a light blue shirt. The background is plain and white, emphasizing the subject."

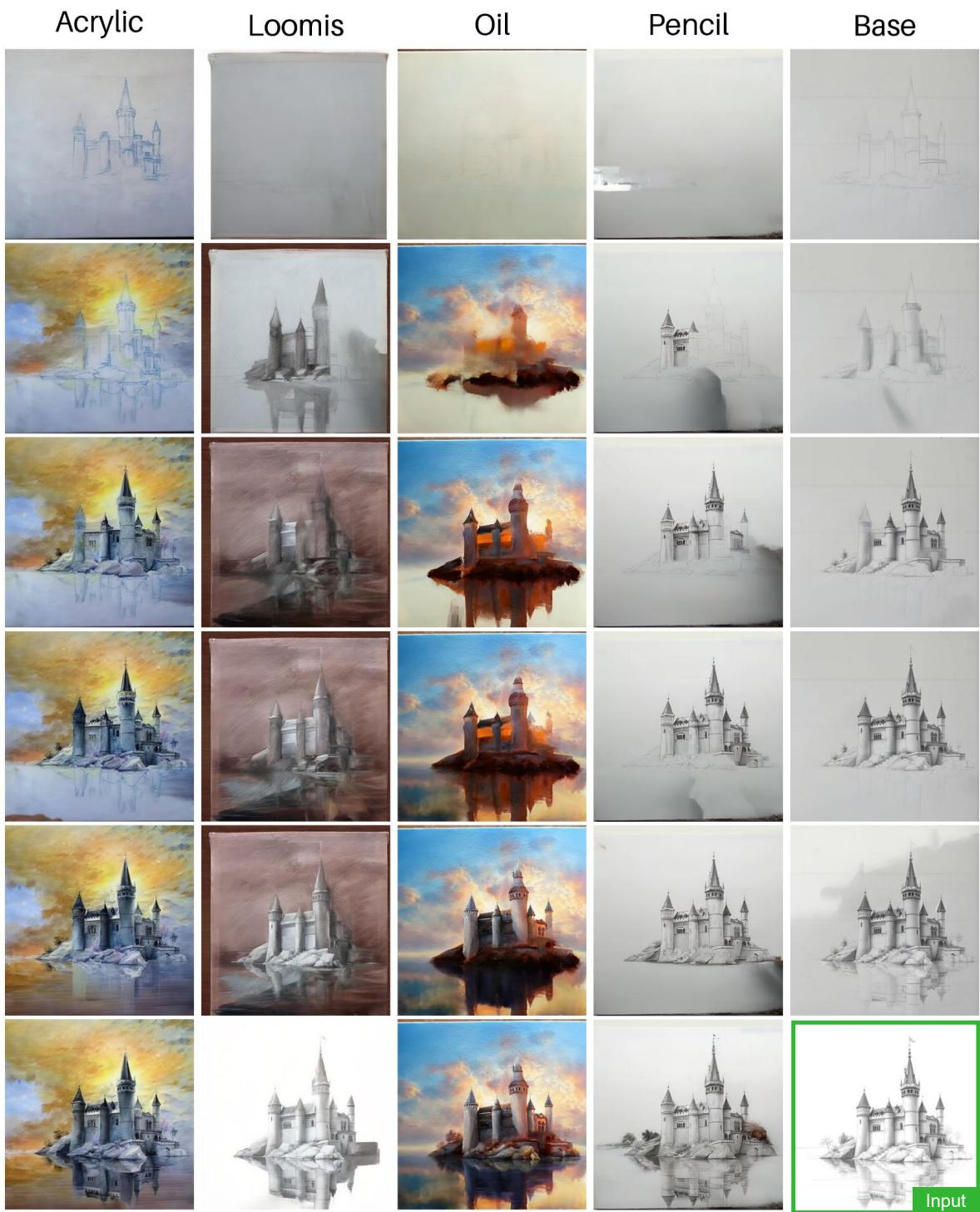


Figure C. Comparison using the same input image (bottom right). Columns 1–4 show linearly sampled outputs from the art media transfer model; column 5 shows the base model output. As the base model's final frame closely matches the input, only the input image is shown. In the case of art media transfer, we used the following prompt with the appropriate art medium inserted: "<art\_media>Step by step painting process. The image depicts a grand castle with multiple towers and turrets, situated on a rocky outcropping overlooking a body of water, with a flag flying atop the central tower."

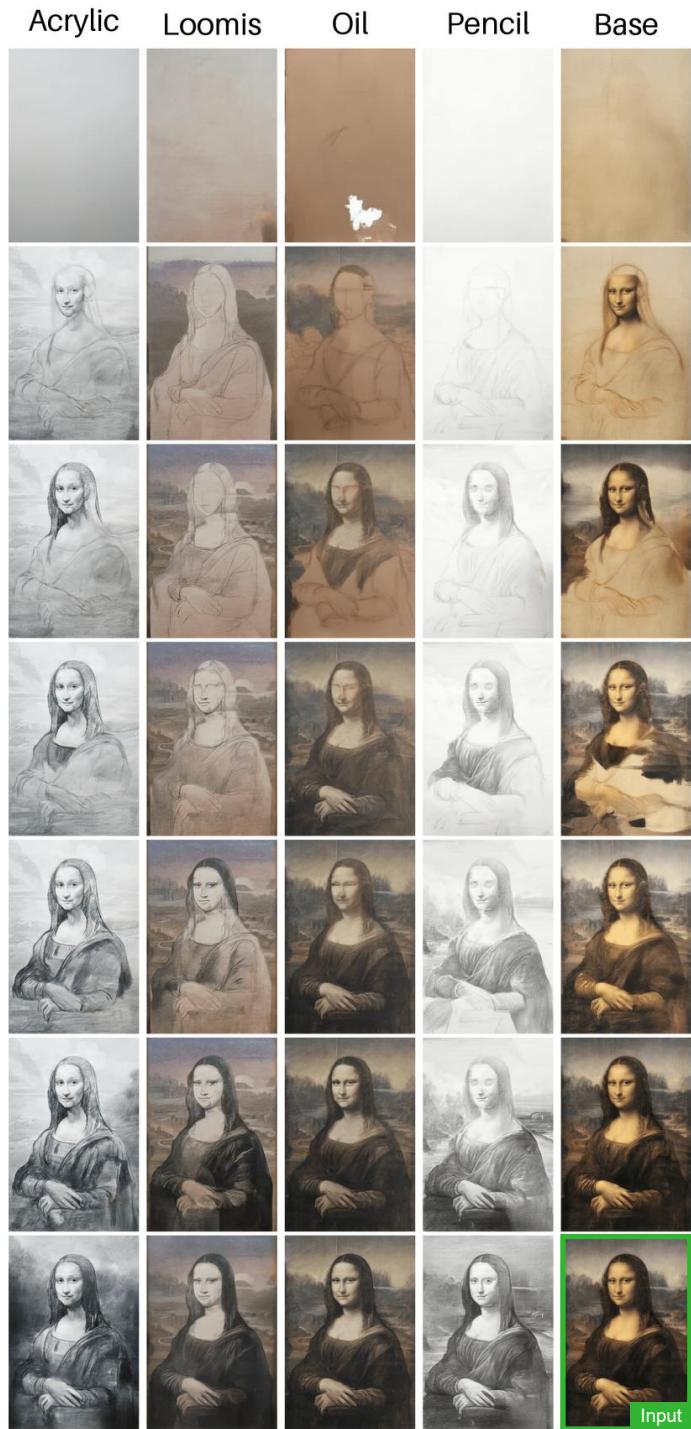


Figure D. Comparison using the same input image (bottom right). Columns 1–4 show linearly sampled outputs from the art media transfer model; column 5 shows the base model output. As the base model’s final frame closely matches the input, only the input image is shown. Mona Lisa by Leonardo da Vinci as input image, image courtesy Wikiart. In the case of art media transfer, we used the following prompt with the appropriate art medium inserted: "<art\_media>Step by step painting process. The image features a woman with long hair, wearing a dark dress with a light collar, set against a background with a body of water and distant mountains."