IMPASTO: Integrating Model-Based Planning with Learned Dynamics Models for Robotic Oil Painting Reproduction

ICRA Submission #1424

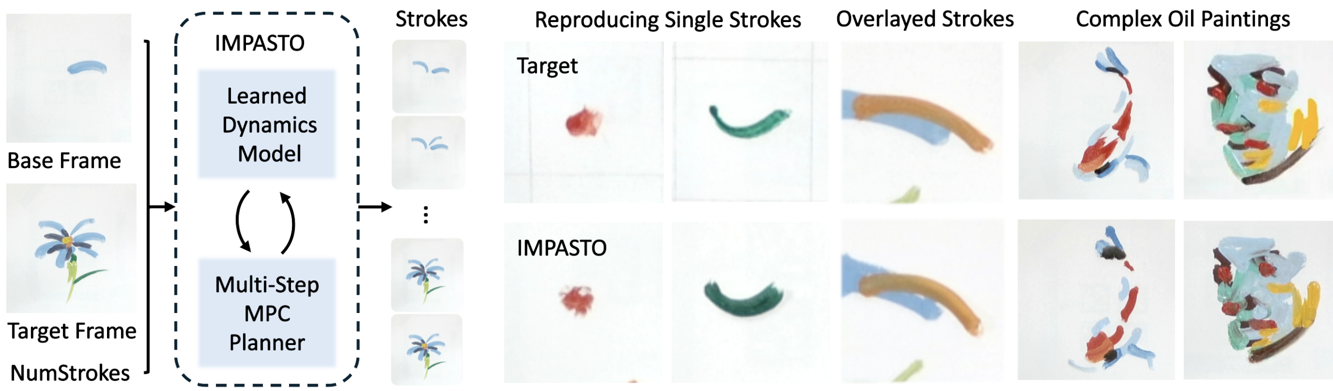

Fig. 1: IMPASTO is a robotic oil painting system that integrates learned neural dynamics models with model-based planning algorithms to accurately replicate human artists' brushstrokes and artworks.

***Abstract*— Robotic reproduction of oil paintings using soft brushes and pigments requires force-sensitive control of deformable tools, prediction of brushstroke effects, and multi-step stroke planning, often without human step-by-step demonstrations or faithful simulators. Given only a target oil painting image, can a robot infer and execute the stroke trajectories, forces, and colors needed to reproduce it? We present IMPASTO, a robotic oil-painting system that integrates learned pixel dynamics models with model-based planning. The dynamics models predict canvas updates from image observations and parameterized stroke actions; a receding-horizon model predictive control optimizer then plans trajectories and forces, while a force-sensitive controller executes strokes on a 7-DoF robot arm. IMPASTO learns solely from robot self-play and achieves high-fidelity replication on human artists' single-stroke datasets and multi-stroke artworks, outperforming baselines in reproduction accuracy. By integrating low-level force control, learned dynamics models, and high-level closed-loop planning, IMPASTO is a step toward robots that can paint with the finesse of a human artist by manipulating real brushes and paints. Project website: `https://impasto-art.github.io/`.**

I. Introduction

Robot painting reproduction—enabling machines to accurately replicate physical artworks with brushes and pigments—has a rich history and presents unique technical challenges [1, 2]. In modern times, these robots often paint images on canvas by executing sequences of brushstrokes and adjusting based on sensory feedback [1, 3, 4]. Unlike digital image generation, a painting robot must master complex low-level control of deformable tools and fluids, force-sensitive manipulation of a brush against a surface, visual perception of the evolving artwork, and high-level planning of stroke sequences. These requirements make robot painting reproduction a significant challenge: the robot must achieve the same nuanced, variable brushstrokes as a human artist based on real-world physics.

We ask: *Given only a static image of an oil painting by an expert artist, can a robot infer the corresponding control actions, such as trajectory, orientation, and applied force, to accurately reproduce the painting?* This setting is common in art training—we hope our robot can reach the level of beginning art students who are able to replicate an oil painting. However, this is a challenging setting for robotics, as it simultaneously 1) requires precise low-level control to replicate dedicated strokes with soft brushes and wet paints; 2) requires high-level planning to compose multiple strokes; 3) assumes no access to any form of action demonstration data, either through teleoperation or tracking the brushes.

We argue that the key to tackling this challenge is *learning a 2D dynamics model*—a robot's internal model for the oil painting domain. Given the current visual state of the canvas, and a parameterized action, the robot should be able to predict the consequence of its brush action, i.e., the resulting state of the canvas. We conjecture that such a model can be trained through self-play. Once trained, the robot can integrate this model with model predictive control (MPC) to infer the correct actions to replicate each single stroke, plan multiple strokes to accomplish a complex painting. Although learning neural dynamics models and combining them with MPC have become increasingly popular in different robotic tasks [5–9], our work is the first attempt to leverage them for robot painting.

Here we present **IMPASTO** (**I**ntegrating **M**odel-Based **P**lanning with Le**A**rned Dynamic**S** Models for Robo**T**ic **O**il Painting Reproduction), a robotic system that combines learned dynamics models with force-sensitive control and MPC-based planning to accurately reproduce human oil paintings. Our contributions include:

- Learned pixel dynamics model: We develop a neural network dynamics model of brushstrokes, trained on a dataset of robot self-play data. This model learns the complex relationship between the robot's action parameters (end-effector trajectory, applied force) and the resulting stroke outcomes, allowing accurate prediction of stroke shape and appearance.

- Model predictive trajectory and force planning: We integrate the learned model into an MPC framework for single or consecutive stroke planning. Given a desired appearance of stroke(s), the MPC optimizer computes precise trajectories and force.
- An integrated robot painting system: We implement a complete system on a 7-DoF robotic arm with a force-torque sensor. The system autonomously handles dipping the brush in paint, cleaning the brush between colors, and painting the strokes on a canvas.

We extensively evaluate IMPASTO on both individual strokes and full paintings drawn by expert human artists. Our results suggest that the learned model plus MPC approach yields strokes that closely match the intended appearances of the human brushstrokes. We demonstrate that the learned dynamics model can better replicate individual and overlaid strokes than the baseline methods. We also show closed-loop painting experiments in which the robot plans and paints a complex multi-stroke artwork.

II. Related Work

A. Modern Robot Painting

The idea of automating art dates back centuries [1]. Harold Cohen's AARON in 1986 is widely considered the first modern robot painter with an open-loop programmed approach [10]. Recently, painting robots have employed closed-loop control. e-David is an industrial robotic arm equipped with a camera that paints with real brushes and continuously adjusts its strategy based on visual feedback [3, 11]. Another state-of-the-art system is the FRIDA series [4, 12, 13], a collaborative painting robot that introduces a differentiable simulation for brush strokes and a real2sim2real planning loop. FRIDA simulates how each stroke will appear and, during execution, periodically captures an image of the canvas to re-plan the remaining strokes, though FRIDA's primary goal was not exact reproduction of existing artworks. These modern robotic painting systems emphasize sophisticated stroke planning algorithms, from high-level image segmentation and stroke sequencing [3] to optimization of stroke parameters in simulation [4], all aimed at achieving human-like painting results under robotic control.

Recent advances in robotic painting leverage machine learning methods. Researchers have applied reinforcement learning (RL) to optimize paint coverage [14, 15] and stroke planning [16–18], used genetic algorithms for stroke planning [19], used imitation learning (IL) to leverage human demonstrations [20, 21], and employed deep generative models to synthesize and stylize brush strokes [22–25]. Together, these approaches push robotic art beyond hand-designed programming toward more autonomous and creative behaviors. It is worth noticing that our problem setting is among the most challenging ones: We aim at exactly replicating human experts' brushstrokes with soft brushes and wet pigments. Unlike in the RL settings, we do not rely on a painting simulator; unlike in the IL setting, we do not have access to human demonstrations.

B. Planning with Learned Neural Dynamics Models

Deep neural networks trained on interaction data have become a standard way to learn dynamics models for manipulation, with broad empirical success [6, 7]. Such models can be trained directly in pixel space [26–30] or in compact latent spaces that abstract observations [8, 31–35]. Beyond these representations, structured scene encodings improve modeling fidelity and generalization, including keypoints [36–39], particles [40–42], and mesh-based parameterizations [9]. For robot painting, a pixel-based dynamics model is a natural choice. Planning over learned dynamics is difficult due to their nonlinearity and nonconvexity. A prevalent strategy is online sampling-based optimization, most commonly cross-entropy methods (CEM) [43] and Model Predictive Path Integral (MPPI) [44], as used by prior work for planning in manipulation tasks [5, 27, 29, 37, 45–47]. These methods require a large number of samples as the action dimension grows. Meanwhile, gradient-based trajectory optimization [40, 48] is prone to local minima and non-smooth objectives.

III. Method

We now describe the parameterization of brushstrokes, the hardware system setup for oil painting, the architecture design and training of the dynamics model, the multi-step planning algorithm, and baseline methods for comparison.

A. Brush Stroke Model

Although there are several possible ways to model brushstrokes [4, 25, 49, 50], oil paintings require curved strokes with precise force control. We parameterize a brushstroke by six parameters, as shown in Fig. 2: the starting location $p_0 = (x_0, y_0)$ in image pixels, the stroke length l, a scalar bend b that curves the stroke up or down, the in-plane orientation α (degrees), normal force F that modulates deposited paint thickness and width, and the paint color $C = (R, G, B, A) \in [0, 1]^4$. Compared to FRIDA [4], we

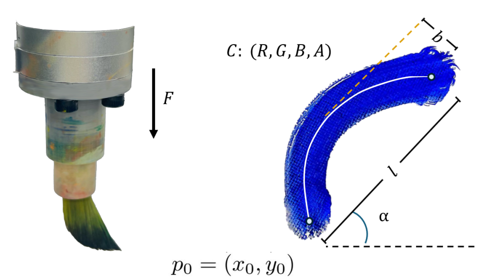

Fig. 2: Action parameterization. Each brushstroke is represented by a quadratic Bézier curve, with a starting location $p_0 = (x_0, y_0)$, length l, bend b, orientation α, force F (controls thickness and width), and color $C \in [0, 1]^4$ (RGBA).

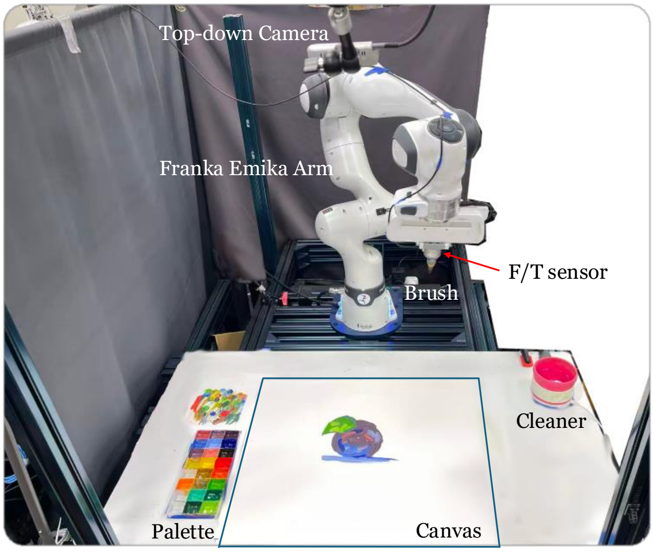

Fig. 3: Hardware system setup of IMPASTO.

replace their parameter h, which specifies how far the brush is pressed to the canvas, with the force parameter F, allowing for force-sensitive control. We denote the action as

$$u = (p_0,\, l,\, b,\, \alpha,\, F,\, C). \tag{1}$$

To map (l, b, α) into a continuous stroke centerline for a minimal but sufficiently representative stroke, we construct a quadratic Bézier curve with control points

$$\mathbf{q}_0 = p_0, \quad \mathbf{q}_1 = p_0 + \tfrac{l}{2}\,\mathbf{t}_\alpha + b\,\mathbf{n}_\alpha, \quad \mathbf{q}_2 = p_0 + l\,\mathbf{t}_\alpha, \tag{2}$$

where $\mathbf{t}_\alpha = [\cos\alpha, \sin\alpha]^\top$ and $\mathbf{n}_\alpha = [-\sin\alpha, \cos\alpha]^\top$. The brush width $w(F)$ is obtained from a calibration curve that monotonically maps normal force to footprint width. We clip u to task-specific bounds $\mathcal{U}$ for learning. The rendered stroke is produced by rasterizing the curve with thickness $w(F)$, then compositing onto the canvas.

B. The Hardware System and Robot Control

To execute brushstrokes on a canvas, we use a 7-DoF Franka Emika Panda equipped with a 6-axis force/torque sensor (Kunwei Tech Inc.) mounted at the wrist, as shown in Fig. 3. The force/torque sensor measures forces from $0.1N$ to $4N$, making it suitable for oil painting. A round brush is rigidly attached to the end effector and held perpendicular to the canvas. We control the arm via the Franka Control Interface in impedance mode and implement a force loop along the surface normal. At the beginning of each stroke, an outer normal-force admittance accumulates a smoothed feedforward force as:

$$F^{\mathrm{ff}}_{z,k+1} = F^{\mathrm{ff}}_{z,k} + k_f\,\mathrm{EMA}_\lambda[F^\star - F_{z,k}]\,\Delta t, \tag{3}$$

where k_f is the feedforward admittance gain, $F^\star$ is the desired normal force set by the stroke action's F, $F_{z,k}$ is the measured normal force at step k, $\mathrm{EMA}_\lambda[\cdot]$ is an exponential moving average with coefficient λ. The controller injects force $\mathbf{F}^{\mathrm{ff}}_b = [0, 0, F^{\mathrm{ff}}_z]^\top$ in the base frame via the Jacobian transpose:

$$\tau = K_p(q_d - q) - K_d\,\dot{q} + s\,J_p^\top\,F^{ff}_{ee}. \tag{4}$$

where s is a scaling factor. The canvas is an 18×21 inch board. Premixed pigments (12 colors) are provided in palette trays; a motorized water spinner cleans the brush between strokes. A top-down Intel RealSense D435 provides RGB observations; images are undistorted, homography-warped to the canvas plane, and center-cropped for learning. Additional details about the hardware setup are in Appendix A.

C. The Pixel Dynamics Model and Training

a) Pixel dynamics model.: The core challenge of this work is to design and train a forward dynamics model for oil painting. Given the current image observation I_t and a candidate stroke action u_t, a differentiable dynamics model predicts the next observation:

$$\hat{I}_{t+1} = f_\theta(I_t, u_t). \tag{5}$$

Architecture-wise, we employ an image encoder ϕ and an action encoder ψ whose embeddings are fused by a decoder φ to output $\hat{I}_{t+1}$. Concretely,

$$z_I = \phi(I_t), \quad z_u = \psi(u_t), \quad \hat{I}_{t+1} = \varphi\big(\mathrm{Concat}(z_I,\, z_u)\big). \tag{6}$$

The neural network architecture is shown in Fig. 4. The encoder-decoder follows U-Net [51]. Inspired by [7], we render stroke actions as an image and feed the rendered image into the action encoder. All the input images of the neural network are cropped around the target stroke and resized to $100 \times 100 \times 1$ grayscale patches. Each action u_t is represented as a 6D vector of $(p_0(x), p_0(y), l, b, F, \alpha)$. The action encoder ψ bridges the modality gap between low-dimensional vector controls and high-dimensional raster states, enabling the network to perceive footprint, orientation, and thickness directly in image space.

We separate color estimation from shape prediction, so the dynamics model focuses on stroke geometry. We run a nearest-neighbor search over a patch database of pigments: inside the stroke-difference mask, we form a representative feature and compare it to patch prototypes using a linear-RGB with Mahalanobis distance [52]. Transparency is decided by a ratio test between base and target intensities.

b) Dataset and training.: The robot self-supervises to collect a dataset $\mathcal{D} = \{(I_t, u_t, I_{t+1})\}$ by uniformly randomly exploring over (p_0, l, b, α, F) while cycling a fixed color set; color-prediction runs use ground-truth color labels. The main objective is a weighted ℓ_1 loss that emphasizes changed pixels near the stroke:

$$\mathcal{L}_{w\ell_1}(\theta) = \frac{1}{\sum_i w_i} \sum_i w_i \left|\hat{I}^{(i)}_{t+1} - I^{(i)}_{t+1}\right|, \tag{7}$$

with w computed from a dilated difference mask between I_t and I_{t+1}. We apply standard data augmentation (random cropping, rotation, and flipping) to augment the dataset. We train f_θ end-to-end with the Adam optimizer. Additional details of the dynamics model design and training can be found in Appendix B.

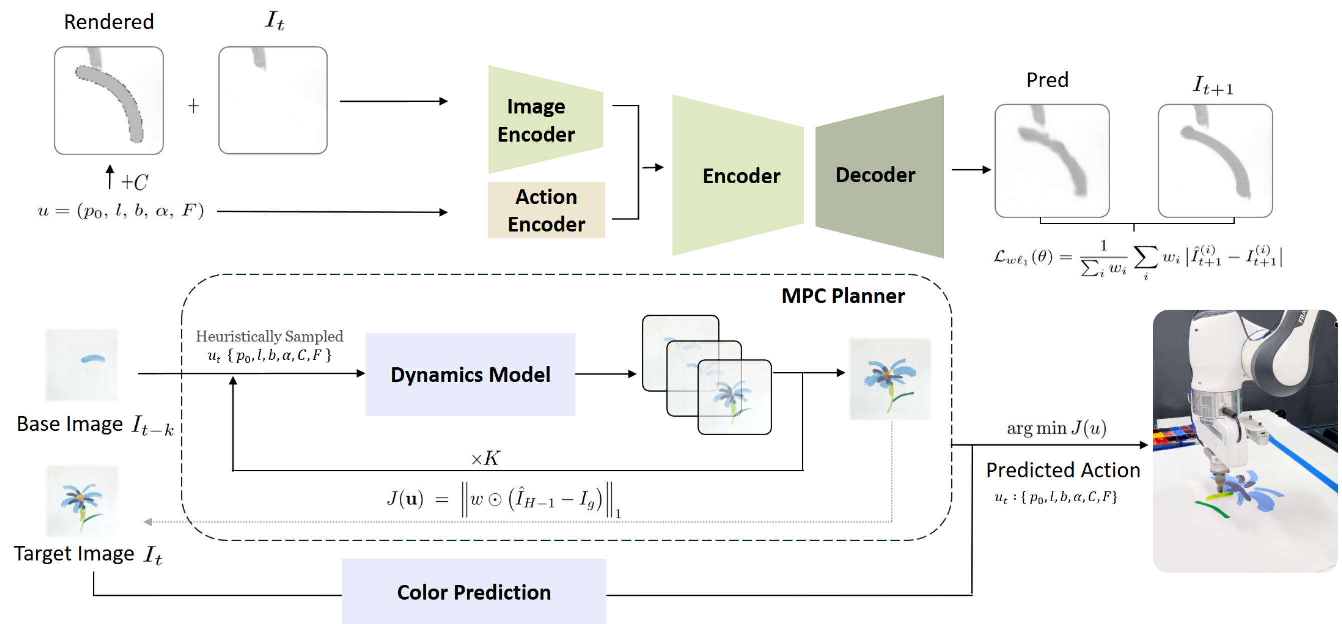

Fig. 4: Overview of the learning and planning framework. Top: IMPASTO-UNet's neural pixel dynamics model, which combines an image encoder and an action encoder to predict the effect of a stroke. The model is trained using a weighted ℓ_1 loss. Bottom: To find one or more consecutive strokes between a base image and a target image, an MPC-based planner optimizes stroke parameters with a weighted ℓ_1 image objective in a receding-horizon, closed loop.

D. Multi-Step Stroke Planning

The learned forward dynamics model allows multi-step planning to accomplish a full painting. At test time, we plan in the action space with receding-horizon model predictive control (MPC). Starting from the current observation I_0 and the goal image I_g, we optimize a length-H sequence $\mathbf{u} = (u_0, \dots, u_{H-1})$ using a weighted image objective:

$$J(\mathbf{u}) = \left\| w \odot \left(\hat{I}_{H-1} - I_g\right) \right\|_1, \quad \text{s.t. } \hat{I}_{t+1} = f_\theta(\hat{I}_t, u_t),\ \hat{I}_0 = I_0 \tag{8}$$

where w up-weights goal-relevant pixels (stroke edges or user-provided masks). We draw K heuristic initializations for each step through a PCA–RANSAC multi-way splitter, perform adaptive sampling and elite-set updates via MPPI [44], clip to $\mathcal{U}$, select the best candidate, execute $u_t^\star$, and re-plan (Alg. 1). This closed-loop, sample-and-refine procedure is robust to model error and naturally incorporates action bounds and force limits. More details can be found in Appendix C.

E. Baseline Methods

We denote our main model as IMPASTO-UNet. Now we introduce several baseline models, which are variants of our method or from related work, including:

IMPASTO-LR: Instead of using a U-Net backbone, this baseline fits a linear regressor that maps $u = (p_0, l, b, \alpha, F)$ directly to an alpha map, which is then composited with the base image and a constant stroke color. This is similar to the method used in Spline-FRIDA [13], where $F \in [0, 1]$ are mapped to a global stroke thickness τ:

$$\tau(F) = \text{softplus}(aF + \beta) + \varepsilon \tag{9}$$

and a parabolic trajectory represented by (p_0, l, b, α) is rasterized by a sequence of spheres with τ as radius. Optionally, a per-pixel darkness field is learned as:

$$\text{Darkness}(x) = \left[\text{clamp}_{[0,1]}\left(1 - \frac{d(x)}{\tau(F)}\right)\right]^c. \tag{10}$$

where $d(x)$ is the Euclidean distance from pixel x to the stroke centerline.

Heuristics-only: For single-step stroke planning, we skeletonize the frame-to-frame difference mask and use the two farthest skeleton endpoints (for p_0 and l), their direction (for α), and the skeleton's maximum signed normal offset (for b), with a fixed force $F = 0.5$.

FRIDA-CNN: FRIDA [4] was not developed to exactly reproduce brushstrokes. This baseline only uses FRIDA's `param2stroke` model, a convolutional neural network predicting a stroke occupancy field from (l, b, F), and is more of a variant of our method. The stroke occupancy is transformed and blended using (p_0, α, C) over the base image.

Additional details of the baselines are in Appendix B. Note that all baseline methods operate *without access to the underlying state context* (i.e., the base image). They *predict only the incremental stroke rather than the next full state image*. To approximate state prediction, we overlay the predicted increment onto the ground-truth base image. In evaluation, this places IMPASTO-UNet at a disadvantage in terms of the weighted ℓ_1 loss, as it must predict the entire next image.

IV. Experiments and Results

Our experiments are designed to address three key questions: $\mathcal{Q}\mathbf{1}$. How much data is needed to train an accurate

Algorithm 1 Multi-step Model-Predictive Control with Adaptive Covariance

Require: Current observation I_0, goal image I_g, dynamics model f_θ, cost c ($L_{w\ell_1}$), candidates K, elite ratio ρ, temperature β, horizon H, EMA `ema`

Ensure: Action sequence $\mathbf{U}^\star = (u_0^\star, \ldots, u_{H-1}^\star)$

$\mathbf{U} \leftarrow \textsc{InitFromMasks}(I_0, I_g)$ ▷ splitter → initial $H\times 6$

for $t = 0, \ldots, H-1$ **do** ▷ MPC recedes, feeds next step

  for iter $= 1, \ldots, M$ **do** ▷ sample-and-refine

    $\mathcal{A} \leftarrow \{\mathbf{U}\}$ ▷ null particle (no noise)

    $\mathcal{A} \leftarrow \mathcal{A} \cup \textsc{SampleAdaptive}(\mathbf{U}, K-|\mathcal{A}|)$

    for all $\tilde{\mathbf{U}} \in \mathcal{A}$ **do**

      $\hat{I}_H \leftarrow \textsc{Rollout}(f_\theta, I_t, \tilde{\mathbf{U}})$

      $J(\tilde{\mathbf{U}}) \leftarrow \|w \odot (\hat{I}_H - I_g)\|_1$ ▷ terminal $L_{w\ell_1}$

    $\mathcal{E} \leftarrow$ top-K_e by J ▷ $K_e = \lfloor \rho K \rfloor$

    $\mathbf{U} \leftarrow \texttt{ema} \cdot \sum_{\tilde{\mathbf{U}} \in \mathcal{E}} \textsc{Softmax}_\beta(-J)\,\tilde{\mathbf{U}} + (1-\texttt{ema}) \cdot \mathbf{U}$

    $(\Sigma_t, \sigma_t)_t \leftarrow \textsc{CMAUpdate}(\mathcal{E})$ ▷ update per-timestep cov/step-size

    $\mathbf{U} \leftarrow \textsc{Clip}$ $\mathbf{U}$ to action bounds

  $I_{t+1} \leftarrow \textsc{Execute}(I_t, u_t^\star)$ ▷ Apply to system / obtain next obs.

return $\mathbf{U}^\star$

dynamics model in our case? $\mathcal{Q}$**2**. Can IMPASTO accurately reproduce expert human artists' strokes? $\mathcal{Q}$**3**. Can IMPASTO enable closed-loop, multi-step planning to accomplish a complex oil painting?

$\mathcal{Q}$1. *IMPASTO's pixel dynamics model can be trained to accurately predict the stroke effects.*

As shown in Fig. 5, as the number of training samples increases, the dynamics model becomes more accurate in predicting the next canvas state. We observe that with a reasonable amount of self-play data ($n = 900$), the model can achieve an accurate prediction with $\mathcal{L}_{\ell_1} = 0.0285$, which was sufficient for the following experiments. We additionally present the fitted lines and the corresponding parameters.

$\mathcal{Q}$2. *IMPASTO can more accurately reproduce human artists' individual strokes than the baseline methods.*

We compare IMPASTO-UNet with the baselines. First, we collected two datasets for evaluation. The first dataset ("Single Strokes") comes from five trained human artists, who are experts in oil painting. Every artist contributed 12 separate brushstrokes that reflected their unique style. The second dataset ("Overlaid Strokes") was produced by a single artist, who freely painted 50 random strokes that were layered on top of each other. Note that these datasets are not used for training.

Then we integrate the learned dynamics model with the multi-step planning algorithm described in Sec. III-D. We first set the planning horizon to be 1 step. We measure the accuracy in two stages: the planning stage and the execution stage. At the planning stage, the 1-step planning algorithm infers the stroke actions, then we use different dynamics models to predict and render the strokes, and then we compare the rendered strokes with ground-truth strokes. At the execution stage, we let the robot execute the stroke action, take a photo of the canvas, and compare that with the ground truth.

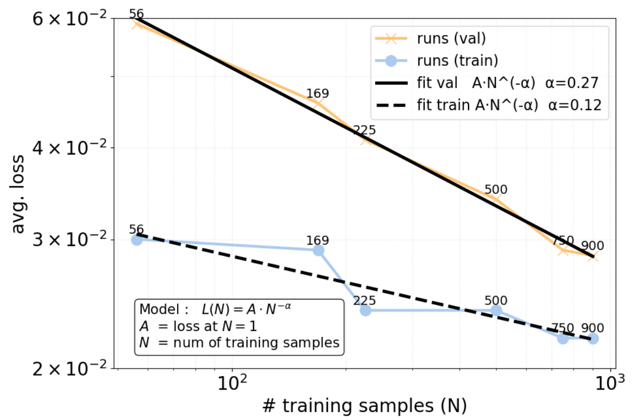

Fig. 5: Training and evaluation loss of IMPASTO's dynamics model vs. number of training samples (log–log scale).

Stage Loss	Planning $\mathcal{L}_{w\ell_1}\downarrow$	Execution $\mathcal{L}_{w\ell_1}\downarrow$	LPIPS↓	Planning $\mathcal{L}_{w\ell_1}\downarrow$	Execution $\mathcal{L}_{w\ell_1}\downarrow$	LPIPS↓
Test Dataset		Artist #1			Artist #2	
IMPASTO-UNet	**0.0197**	**0.0506**	**0.1874**	**0.0225**	**0.0413**	**0.1675**
FRIDA-CNN	0.0215	0.0617	0.2736	0.0318	0.0508	0.2348
IMPASTO-LR	0.0229	0.0718	0.2511	0.0242	0.0434	0.1711
Heuristics-only	N/A	0.0541	0.2001	N/A	0.0626	0.2001
Test Dataset		Artist #3			Artist #4	
IMPASTO-UNet	0.0257	0.0745	**0.2236**	**0.0234**	0.0346	**0.1621**
FRIDA-CNN	0.0278	0.1062	0.3495	0.0256	0.0362	0.2623
IMPASTO-LR	**0.0202**	**0.0721**	0.2622	0.0271	**0.0331**	0.2338
Heuristics-only	N/A	0.0834	0.2455	N/A	0.0556	0.2246
Test Dataset		Artist #5			Overlaid	
IMPASTO-UNet	**0.0199**	0.0457	**0.1633**	**0.0258**	**0.0907**	**0.2209**
FRIDA-CNN	0.0264	0.0515	0.2781	0.0288	0.1034	0.2645
IMPASTO-LR	0.0230	**0.0402**	0.1869	0.0263	0.0929	0.2473
Heuristics-only	N/A	0.0574	0.2017	N/A	0.0940	0.2554

TABLE I: Learned dynamics models' planning and execution performance in terms of $\mathcal{L}_{w\ell_1}$ and LPIPS losses.

We use two metrics for evaluation. The first one was the weighted ℓ_1 loss used as the objective in learning and planning. The second one was the popular LPIPS (Learned Perceptual Image Patch Similarity) metric [53], which is a perceptual metric that measures the similarity between two images based on deep neural network feature embeddings, aligning better with human visual judgments than pixel-wise differences. This metric is particularly useful for the execution stage due to the differences in canvas color and lighting conditions.

The results are shown in Table I. The visualization of results can be found in Fig. 6 and Fig. 7. Our U-Net-based model consistently performs better than baselines in terms of LPIPS metrics in all cases, although the linear regression variant performs slightly better in terms of the weighted ℓ_1 loss for some artists. On average across all artists' data, IMPASTO-UNet reduces planning $\mathcal{L}_{w\ell_1}$ loss by 16.45% vs. FRIDA-CNN and 5.28% vs. IMPASTO-LR; execution $\mathcal{L}_{w\ell_1}$ loss by 19.48%, 5.33%, and 21.21% vs. FRIDA-CNN,

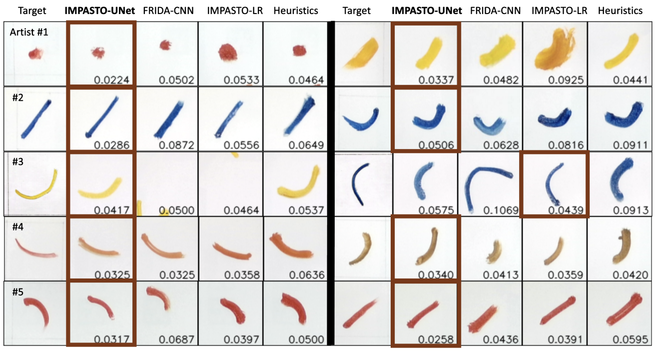

Fig. 6: Target brushstrokes from five human artists (two examples per artist) and the strokes reproduced by the robot using different methods. The numbers shown are the weighted ℓ_1 loss between the target and the painted strokes. Instances with the best performances are highlighted with bold borders. Overall, IMPASTO-UNet more accurately reproduced human brushstrokes.

IMPASTO-LR, and Heuristics-only; and LPIPS by 35.36%, 18.21%, and 15.68%, respectively. Notably, although our dynamics model was trained using self-supervised play data, it performed reasonably well in reproducing brushstrokes from different human artists.

IMPASTO also performs better than all the baselines in the "Overlaid" dataset, suggesting that it can make accurate predictions given noisy base images. IMPASTO-UNet reduces planning $\mathcal{L}_{w\ell_1}$ loss by 10.42% vs. FRIDA-CNN and 1.90% vs. IMPASTO-LR; execution $\mathcal{L}_{w\ell_1}$ loss 12.28%, 2.37%, and 3.51% vs. FRIDA-CNN, IMPASTO-LR, and Heuristics-only; and LPIPS 16.48%, 10.68%, and 13.51%.

$\mathcal{Q}$3. IMPASTO enables closed-loop, multi-step planning to replicate complex oil paintings.

Next, we show that IMPASTO can integrate long-horizon, multi-step planning to replicate oil paintings with many strokes. We first ask an artist to paint two oil paintings ("Flower" and "Fish"), consisting of 17 strokes and 18 strokes, respectively. We then set the planning horizon to be 5 steps, i.e., step 5, 10, 15, and the final images are used as target images. We compare IMPASTO with FRIDA-CNN, the best performing baseline other than IMPASTO's own variant, using the same metrics. The quantitative results are shown in Table II, and the visualizations are in Fig. 8. IMPASTO performs better in multi-step planning, thanks to its higher accuracy in reproducing individual brushstrokes. On average, IMPASTO-UNet reduces planning $\mathcal{L}_{w\ell_1}$ loss by 24.01% vs. FRIDA-CNN; execution $\mathcal{L}_{w\ell_1}$ loss by 10.55% vs. FRIDA-CNN; and LPIPS by 14.63%.

Finally, as shown in Fig. 9, IMPASTO was able to replicate a rather complex painting with high precision, which requires 40 steps. More examples and the robot painting process can be found in Appendix D and the supplemental video.

V. Conclusions

An oil painting robot must address fine-grained control of deformable tools and fluid dynamics, execute force-sensitive brush-surface interactions, perceive the evolving canvas state, and solve high-level planning of stroke sequences. These requirements make robotic oil painting a inspiring and ambitious challenge for robotics research. We present IMPASTO, a robot painting system that integrates learned dynamics models and model-based planning to replicate oil paintings. We demonstrate that IMPASTO can accurately replicate human artists' brushstrokes and artworks. Through these contributions, we advance robotic painting by fusing learning and control: the robot gains an understanding of brush dynamics from data and uses it in a principled planning framework. IMPASTO represents a step toward robots that can paint with the finesse of a human artist, manipulating real brushes and paints. By integrating low-level force control, visual feedback, learned dynamics, and high-level planning, we move closer to the goal of a robot that can not only reproduce a target image, but do so in a manner capturing the expressivity and precision of human painting.

Test Data	Flower			Fish		
Stage	Planning	Execution		Planning	Execution	
Loss	$\mathcal{L}_{w\ell_1} \downarrow$	$\mathcal{L}_{w\ell_1} \downarrow$	LPIPS↓	$\mathcal{L}_{w\ell_1} \downarrow$	$\mathcal{L}_{w\ell_1} \downarrow$	LPIPS↓
IMPASTO-UNet	**0.0427**	**0.0464**	**0.1293**	**0.0421**	**0.0452**	**0.1117**
FRIDA-CNN	0.0597	0.0501	0.1619	0.0519	0.0523	0.1204

TABLE II: Multi-step stroke planning and execution performance. Planning horizon = 5.

References

[1] L. Scalera, A. Gasparetto, S. Seriani, and P. Gallina, "History of drawing robots," in *International Symposium on History of Machines and Mechanisms*. Springer, 2024, pp. 3–17.

[2] A. Karimov, E. Kopets, S. Leonov, L. Scalera, and D. Butusov, "A robot for artistic painting in authentic colors," *Journal of Intelligent & Robotic Systems*, vol. 107, no. 3, p. 34, 2023.

[3] T. Lindemeier, S. Pirk, and O. Deussen, "Image stylization with a painting machine using semantic hints," *Computers & Graphics*, vol. 37, no. 5, pp. 293–301, 2013.

[4] P. Schaldenbrand, J. McCann, and J. Oh, "Frida: A collaborative robot painter with a differentiable, real2sim2real planning environment," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 712–11 718.

[5] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2786–2793.

[6] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu, "Robocook: Long-horizon elasto-plastic object manipulation with diverse tools," 2023.

[7] Y. Wang, Y. Li, K. Driggs-Campbell, L. Fei-Fei, and J. Wu, "Dynamic-resolution model learning for object pile manipulation," in *RSS*, 2023.

[8] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *CoRL*, 2023.

[9] Z. Huang, X. Lin, and D. Held, "Mesh-based dynamics model with occlusion reasoning for cloth manipulation," in *RSS*, 2022.

[10] H. Cohen, "The further exploits of aaron, painter," *Stanford Humanities Review*, vol. 4, no. 2, pp. 141–158, 1995.

[11] O. Deussen, T. Lindemeier, S. Pirk, and M. Tautzenberger, "Feedback-guided stroke placement for a painting machine," in *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, 2012, pp. 25–33.

[12] P. Schaldenbrand, G. Parmar, J.-Y. Zhu, J. McCann, and J. Oh, "Cofrida: Self-supervised fine-tuning for human-robot co-painting," in *ICRA*. IEEE, 2024, pp. 2296–2302.

[13] L. Chen, P. Schaldenbrand, T. Shankar, L. Coleman, and J. Oh, "Spline-frida: Towards diverse, humanlike robot painting styles with a sample-efficient, differentiable brush stroke model," *IEEE RAL*, 2025.

[14] E. Berrocal, I. Merino, A. F. Montaño, and B. Sierra, "Reinforcement learning for autonomous surface painting with a robotic arm," *Available at SSRN 5396199*.

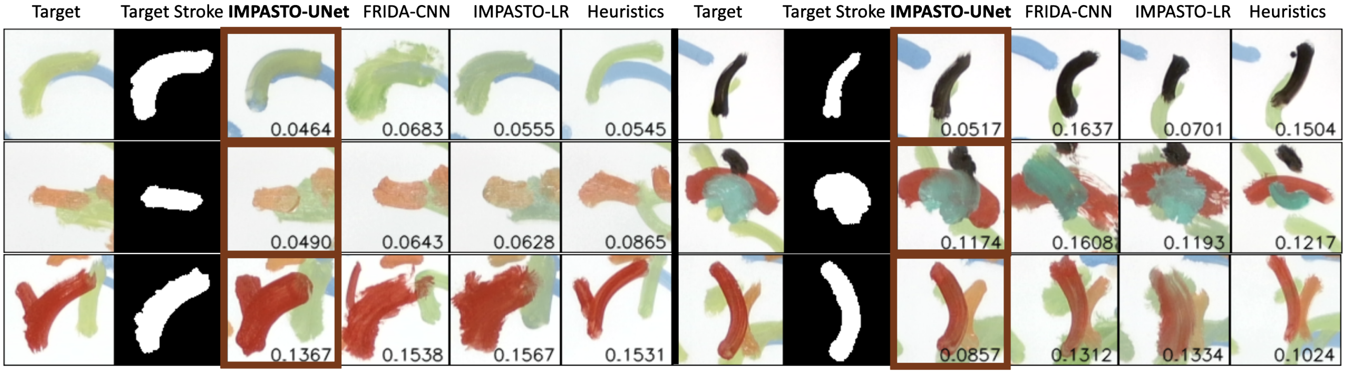

Fig. 7: Qualitative results showing the target strokes from overlaid strokes and the strokes painted by the robot using different methods. Instances with the best performances are highlighted with bold borders. The difference with Fig. 6 is that the base images (canvas states) already have painted strokes. This requires the dynamics models to make accurate predictions given the noisy background. The numbers shown are the weighted ℓ_1 loss. Note that the loss is *only* calculated around the target stroke area. IMPASTO-UNet is more accurate in reproducing human brushstrokes given the noisy base images.

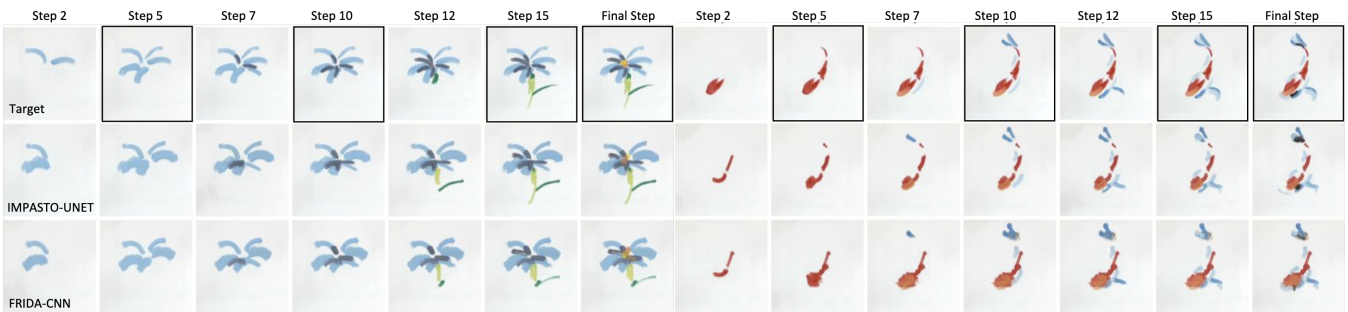

Fig. 8: Qualitative results showing the target paintings and the painting produced by the robot using IMPASTO vs. FRIDA. The planning horizon was set to be five, and the framed images are the targets for MPC. Our method can more accurately reproduce oil paintings, as shown by quantitative results in Table II.

[15] G. Tiboni, R. Camoriano, and T. Tommasi, "Paintnet: Unstructured multi-path learning from 3d point clouds for robotic spray painting," in *2023 IROS*. IEEE, 2023, pp. 3857–3864.

[16] P. Schaldenbrand and J. Oh, "Content masked loss: Human-like brush stroke planning in a reinforcement learning painting agent," in *AAAI*, 2021.

[17] G. Lee, M. Kim, M. Lee, and B.-T. Zhang, "From scratch to sketch: Deep decoupled hierarchical reinforcement learning for robotic sketching agent," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5553–5559.

[18] B. Jia and D. Manocha, "Sim-to-real brush manipulation using behavior cloning and reinforcement learning," *arXiv preprint arXiv:2309.08457*, 2023.

[19] C. Aguilar and H. Lipson, "A robotic system for interpreting images into painted artwork," in *International conference on generative art*, vol. 11, 2008.

[20] Y. Park, S. Jeon, and T. Lee, "Robot learning to paint from demonstrations," in *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2022, pp. 3053–3060.

[21] C. Guo, T. Bai, X. Wang, X. Zhang, Y. Lu, X. Dai, and F.-Y. Wang, "Shadowpainter: Active learning enabled robotic painting through visual measurement and reproduction of the artistic creation process," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 3, p. 61, 2022.

[22] R. Wu, Z. Chen, Z. Wang, J. Yang, and S. Marschner, "Brush stroke synthesis with a generative adversarial network driven by physically based simulation," in *Proceedings of the joint symposium on computational aesthetics and sketch-based interfaces and modeling and non-photorealistic animation and rendering*, 2018, pp. 1–10.

[23] S. Liu, T. Lin, D. He, F. Li, R. Deng, X. Li, E. Ding, and H. Wang, "Paint transformer: Feed forward neural painting with stroke prediction," in *ICCV*, 2021, pp. 6598–6607.

[24] J. M. Gülzow, P. Paetzold, and O. Deussen, "Recent developments regarding painting robots for research in automatic painting, artificial creativity, and machine learning," *Applied Sciences*, vol. 10, no. 10, p. 3396, 2020.

[25] D. Guo and G. Yan, "B-bsmg: Bézier brush stroke model-based generator for robotic chinese calligraphy," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, p. 104, 2024.

[26] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," *arXiv preprint arXiv:1605.07157*, 2016.

[27] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections." in *CoRL*, 2017.

[28] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[29] L. Yen-Chen, M. Bauza, and P. Isola, "Experience-embedded visual foresight," in *CoRL*. PMLR, 2020, pp. 1015–1024.

[30] H. Suh and R. Tedrake, "The surprising effectiveness of linear models for visual foresight in object pile manipulation," *arXiv preprint arXiv:2002.09093*, 2020.

[31] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," *arXiv preprint arXiv:1506.07365*, 2015.

[32] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," *arXiv preprint arXiv:1606.07419*, 2016.

[33] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[34] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*, 2019, pp. 2555–2565.

[35] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel *et al.*,

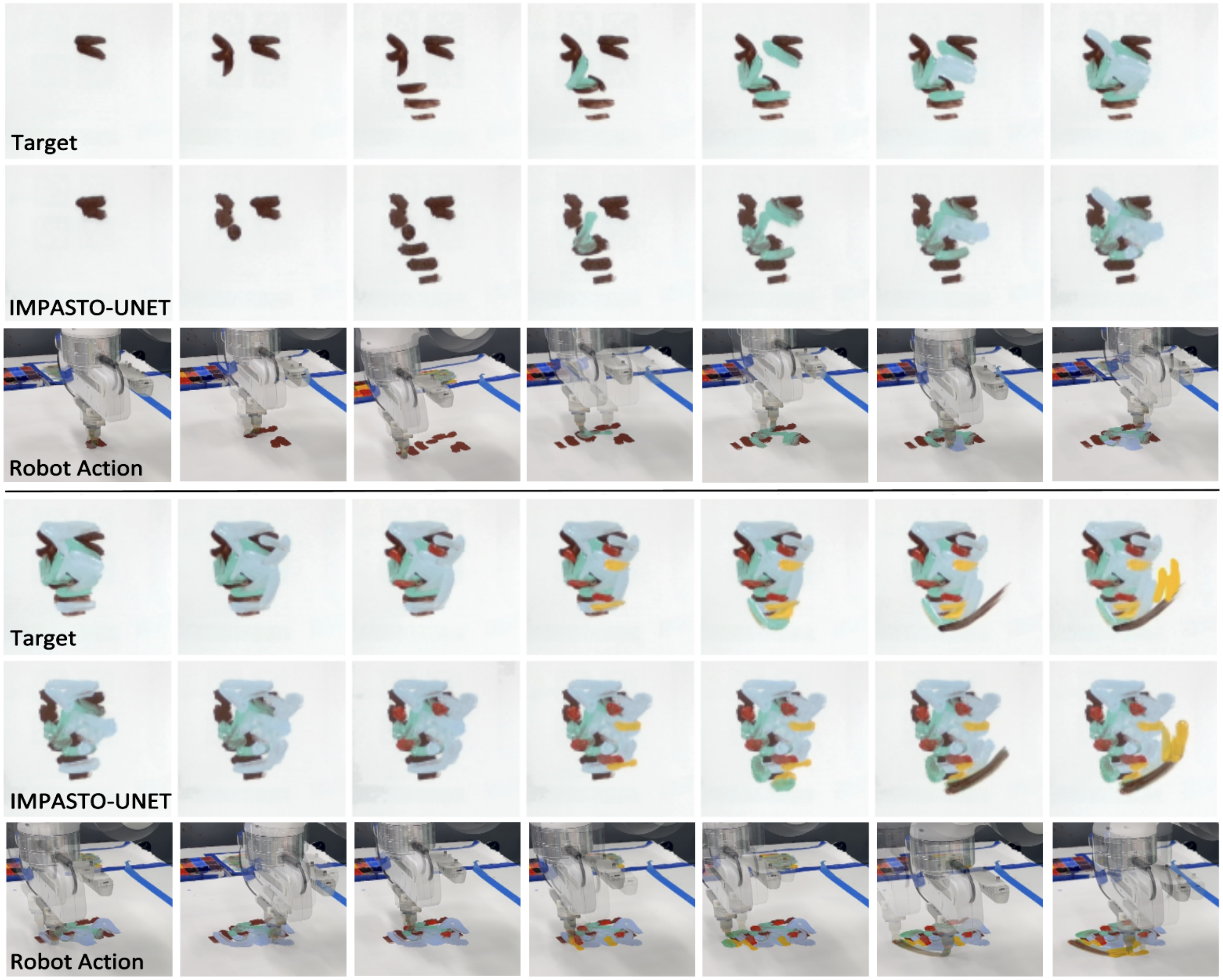

Fig. 9: IMPASTO was able to replicate a complex oil painting with 40 strokes (not all steps are shown).

"Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[36] T. D. Kulkarni, A. Gupta, C. Ionescu, S. Borgeaud, M. Reynolds, A. Zisserman, and V. Mnih, "Unsupervised learning of object keypoints for perception and control," *Advances in neural information processing systems*, vol. 32, pp. 10 724–10 734, 2019.

[37] L. Manuelli, Y. Li, P. Florence, and R. Tedrake, "Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning," *arXiv preprint arXiv:2009.05085*, 2020.

[38] Y. Li, A. Torralba, A. Anandkumar, D. Fox, and A. Garg, "Causal discovery in physical systems from videos," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[39] K. Shen, J. Yu, J. Barreiros, H. Zhang, and Y. Li, "Bab-nd: Long-horizon motion planning with branch-and-bound and neural dynamics," in *The Thirteenth ICLR*.

[40] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," *arXiv preprint arXiv:1810.01566*, 2018.

[41] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu, "Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks," *arXiv preprint arXiv:2205.02909*, 2022.

[42] K. Zhang, B. Li, K. Hauser, and Y. Li, "Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[43] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[44] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, pp. 344–357, 2017.

[45] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," in *CoRL*, 2020.

[46] J. Sacks, R. Rana, K. Huang, A. Spitzer, G. Shi, and B. Boots, "Deep model predictive optimization," 2023.

[47] T. Han, A. Liu, A. Li, A. Spitzer, G. Shi, and B. Boots, "Model predictive control for aggressive driving over uneven terrain," 2024.

[48] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, "Propagation networks for model-based control under partial observation," in *ICRA*. IEEE, 2019, pp. 1205–1211.

[49] S. Wang, J. Chen, X. Deng, S. Hutchinson, and F. Dellaert, "Robot calligraphy using pseudospectral optimal control in conjunction with a novel dynamic brush model," in *IROS*. IEEE, 2020, pp. 6696–6703.

[50] T.-M. Li, M. Lukáč, M. Gharbi, and J. Ragan-Kelley, "Differentiable vector graphics rasterization for editing and learning," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–15, 2020.

[51] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[52] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[53] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.