Dynamics-Aware Trajectory Generation for Artistic Painting using Diffusion

Gerry Chen, Frank Dellaert, and Seth Hutchinson
Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, USA
{gerry, seth}@gatech.edu, dellaert@cc.gatech.edu

***Abstract*—In this work, we seek to generate robot trajectories for artistic painting which exploit the dynamics unique to a robot embodiment. Denoising Diffusion Probabilistic Models (DDPM) have been shown to be effective at generating not only images, but also many other continuous signals including robot trajectories and stroke-based drawing paths. While existing works generating stroke-based art using DDPMs produce *computer* renderings of drawings, many roboticists and artists have previously identified the value in creating physical artwork with an embodied AI. One notable quality of artwork is the particularities of the medium and tools used. Therefore, we seek to combine artistic stroke generation and dynamics-aware trajectory generation using DDPM to generate strokes that capture both the artistic qualities of the training data and of the robot embodiment. We compare several approaches to extending stroke generation DDPMs to respect robot dynamics, including alternative parameterizations, training on modified data, classifier guidance, and classifier-free guidance. We qualitatively show that classifier-free guidance most effectively exploits the robot embodiment to generate visually pleasing yet dynamically feasible painting trajectories.**

I. Introduction

Robot art presents an ideal task for exploring human-robot relationships. Creating art with robot assistance is inherently a collaborative process between a human artist and robot in negotiating a result that reflects the artist's intent while respecting the robots strengths and weaknesses. It also generates physical artistic artifacts as testaments to the cooperation, which can reach different audiences in an emotional, visual, and intuitive way that may be more accessible than traditional scientific or engineering products. As exploring the qualities specific to the artistic medium and tools (*i.e.* the robot painter in our case) is both a core value of art and an excellent way to express a robot's strengths and limitations, generating artistic painting trajectories which are tailored to a particular robot embodiment is a valuable direction for human-robot interaction research.

Diffusion models have produced amazing advancements in collaborative art generation in recent years. DDPMs for image generation [12, 23] have enabled unprecedented abilities to generate high-quality visual art with low effort, and efforts in video generation [27], editing [15, 16, 23], consistency [15, 9], and multi-modal LLMs [17, 18] have made the process as collaborative and accessible as ever. Nevertheless, they remain limited to exploring the space of computer renderings and do not cross the boundary into human-robot interaction. DDPMs have also been shown to apply well to robot trajectory generation [13, 1, 5] and artistic stroke generation [30, 21] separately, but haven't studied the intersection of the two.

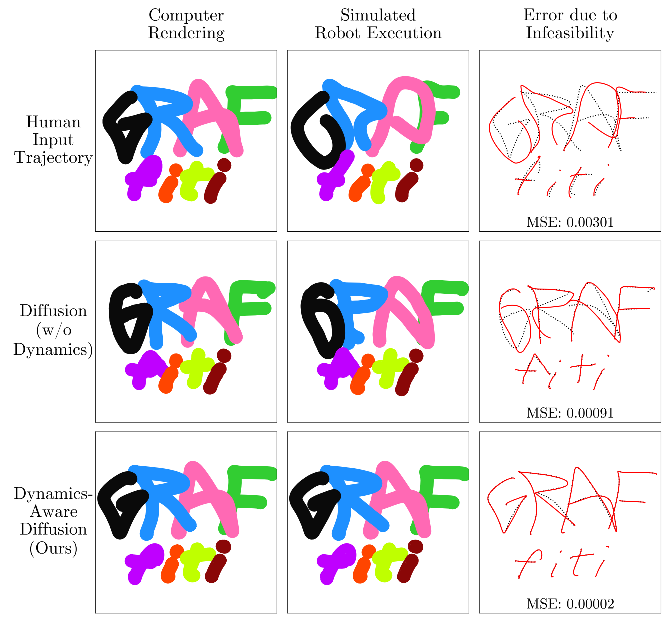

Fig. 1. As an example application, editing trajectories using our DDPM enables augmenting human input trajectories to be dynamically feasible while retaining the graffiti style. Simulating robot execution of raw human inputs (top), edits by diffusion w/o dynamics constraints (middle), and edits by dynamics-aware diffusion using our Approach 3 (bottom) demonstrate the importance of incorporating dynamics to ensure faithful robot execution.

Meanwhile, the field of robot painting has yet to leverage the advantages unlocked by recent advances in DDPMs. Painterly rendering [11] creates painting strokes but needs a target image to recreate. The advent of CLIP [22] and diffvg [14] brought the ability to optimize vector strokes suitable for painting according to textual descriptions thanks to a fully differentiable renderer and loss function [24, 25, 26, 28, 29], but the explicit image rendering somewhat limits the ability to explore the space of possible paintings. They also tend to struggle creating longer strokes. Sketch-RNN and its variants [10, 7] are auto-regressive approaches with a VAE-RNN architecture, but do not exhibit the inference-time flexibility nor the long-range relational capabilities of DDPMs. GAN-based approaches [8] are also popular but suffer similar challenges. Finally, while many works have custom brush models to exploit the unique qualities of the brush painting medium, few if any works have explored how coupling the artistic generation process with the *robot dynamics* can lead to new and interesting artistic results.

In this paper, we present a DDPM-based system for generating artistic painting trajectories tailored to a particular a robot painter. We train an unconditioned DDPM on the #000000book graffiti drawings dataset, then investigate various approaches for tailoring the DDPM to our particular cable robot embodiment. Our contributions include:

- validating that DDPM trajectory generation (*i.e.* Diffuser [13], SketchKnitter [30]) can generalize from robot trajectories and doodles to *graffiti-style* painting strokes;
- identifying that coupling art generation and robot dynamics offers artistic value and HRI benefits; and
- investigating several methods for combining artistic generation with trajectory optimization using DDPMs.

II. Approach

In this section, we first describe our DDPM model for generating graffiti trajectories, then present various approaches to tailoring the model to our robot embodiment.

A. DDPM for Graffiti Trajectory Generation

We base our non-robot-specific DDPM on Diffuser [13] and SketchKnitter [30]. As our goal is just to study combining artistic generation with the robot dynamics and not user input modalities, we do not employ conditioning inputs such as class, text, or images. We use a 1D U-Net architecture (depth-3, kernel size 5, 100 diffusion steps) and generate the state and action sequences together in the form of a 5xN vector containing $[x, y, \Delta x, \Delta y, PenUp]$. We train the model with a selection of 3000 drawings (of about 72000) from the #000000book, centered and normalized (while preserving aspect ratio) and training on random chunks of 512 points within drawings to convergence. Thanks to the 1D U-Net architecture, any-length trajectory can be generated, although there is no global context.

B. Tailoring to Robot Dynamics

We rely on trajectory retiming [4, 19, 6] to create dynamically feasible trajectories (*i.e.* w/ timestamps) for pre-existing painting paths (*i.e.* w/o timestamps). Specifically, any path can be turned into a dynamically feasible one by scaling the artwork to the workspace size then applying retiming to find a dynamically feasible speed profile to follow the path. We compare several approaches to generating painting trajectories tailored to our robot dynamics.

Baseline Approach – Decoupled Generation: As a baseline, we first consider a decoupled approach where we first generate an artistic path using the base, unconditioned model from II-A, then apply retiming to make it dynamically feasible. This approach is simple and easy to implement, but the resulting painting does not clearly reflect the unique dynamics of the robot since the painting was designed without any consideration for the robot.

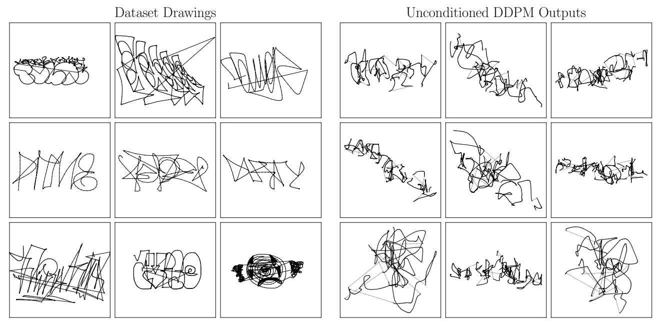

Fig. 2. Output examples (bottom) of the DDPM trained by the #000000book graffiti trajectories dataset (top). Note that we should expect the DDPM only to match the "textures" of the dataset drawings but not the compositions since we focus only on the combination of art generation and robot dynamics, and thus do not give any global context signals.

Approach 1 – Training on Dynamically Feasible Data: In this approach, we first apply trajectory retiming to each drawing in the training set, then train the DDPM on the *dynamically feasible trajectory* data. The resulting DDPM is expected to generate trajectories that reflect the training data's adherence to the robot dynamics.

Approach 2 – Diffusion in the Control Space: In this approach, we train the DDPM on the robot's controls directly, which typically have much simpler boundaries than in the task space or the higher-level action space commonly target by generative models (*e.g.* x/y/pen-up waypoints).

Approach 3 – Classifier-Guided Diffusion: In this approach, we apply inference-time guidance based on the robot constraints. Specifically, we (differentiably) compute the joint controls needed to execute the path (assume constant time interval between consecutive points) at each diffusion step and use the constraint violation as a loss term to guide the diffusion process.

Approach 4 – Classifier-Free Guidance: In this approach, we train the DDPM on both retimed and non-retimed data, with a conditioning input that indicates whether the data is dynamically feasible. During inference time, we supply the appropriate conditioning input (potentially scaled) to generate dynamically feasible trajectories.

III. Preliminary Results

A. DDPM for Graffiti Trajectory Generation

Fig. 2 shows the output of the DDPM trained on the #000000book dataset. Because the DDPM is unconditioned and uses a U-Net architecture (which only has a limited receptive field of only about 40 timesteps), we should only expect the outputs to match the "textures" of the dataset drawings but not the compositional aspects.

B. Tailoring to Robot Dynamics

In these preliminary experiments, we apply the simple dynamics constraints of velocity and acceleration limits. We define trajectory waypoints to be spaced at 50Hz (0.02s per waypoint) and enforce maximum velocities and accelerations

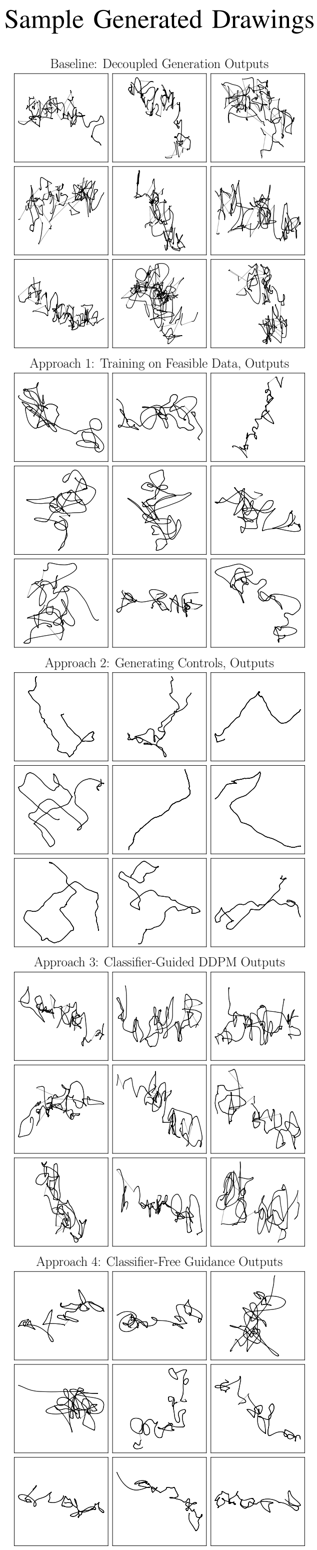

Fig. 3. Output examples of the various approaches to tailoring the DDPM to the robot dynamics. The baseline approach (top) generates paths without considering the robot dynamics, while the other approaches generate paths that respect the robot dynamics.

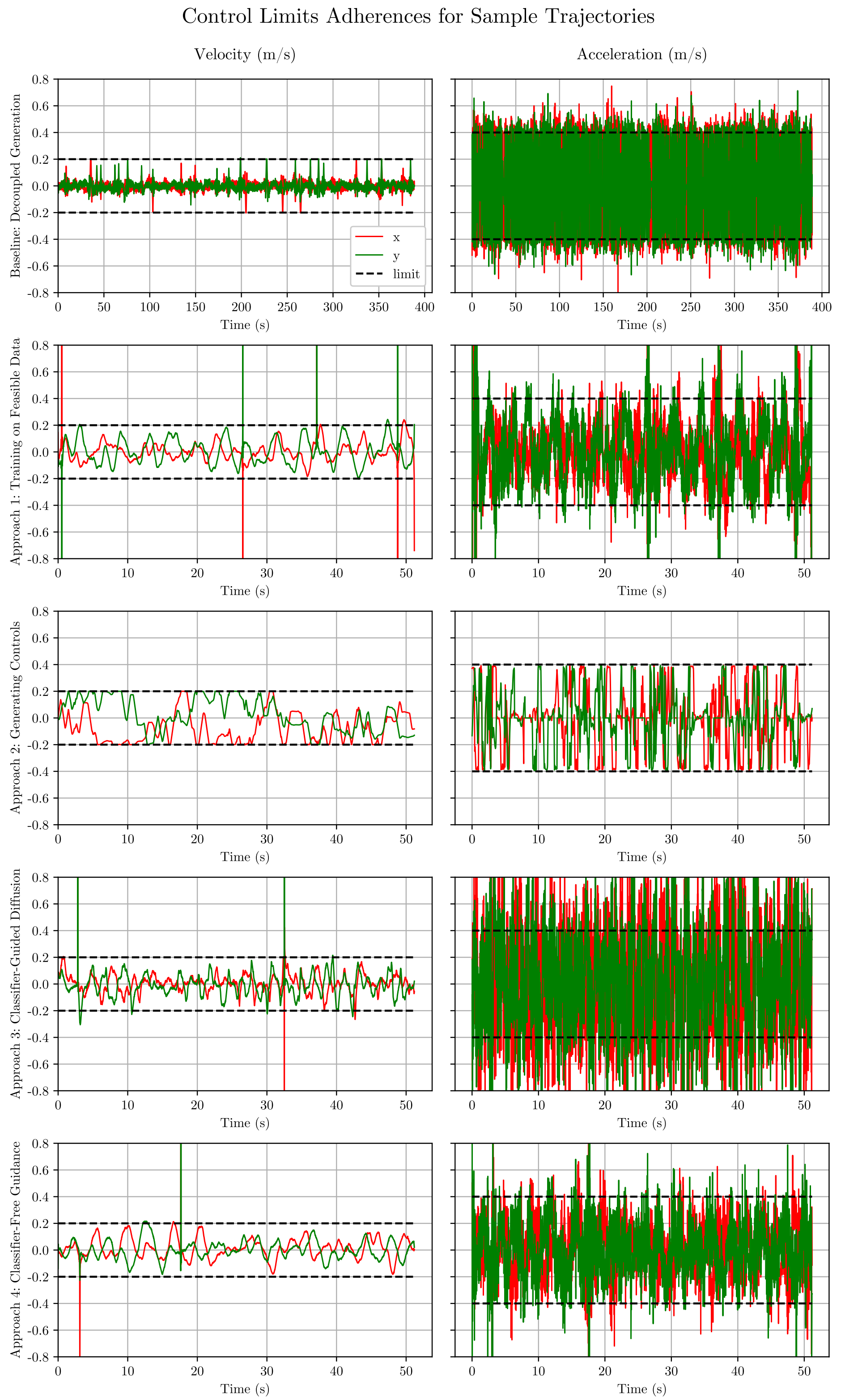

Fig. 4. Velocity and acceleration profiles for sample paths show that most approaches violate the constraints a little bit. Approaches 2 and Baseline (parameterizing output as feasible region and running retiming after generation, respectively) have the least violation (theoretically we expect them to have no violation), while Approach 3 (classifier-guided diffusion) has the most. Single-timestep vertical spikes in velocity profiles are pen-lifts and do not count as violations.

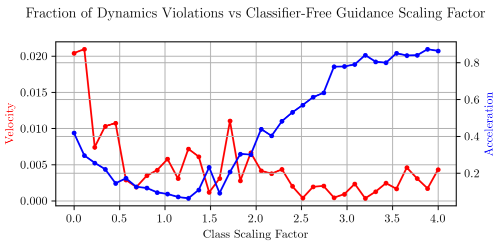

Fig. 5. The scaling factor used for classifier-free guidance conditions the network to output trajectories more in the direction of the retimed (no constraint violation) training trajectories. A scaling factor of 1.3 was used in Figs. 3 and 4.

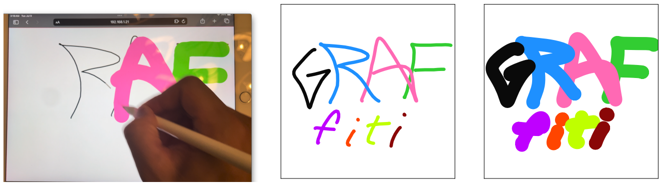

Fig. 6. Trajectory editing application demo allows the user to interactively create AI-assisted robot painting trajectories. The user draws a trajectory (left, thin black lines) while the diffusion model modifies it in real-time (left, thick colored lines) to be both dynamically feasible and stylistically similar to the training data (#000000book, Fig. 2 left). In this example, the input trajectory (center) is modified using a DDPM into a dynamically feasible, stylized trajectory (right) reflecting the training data distribution.

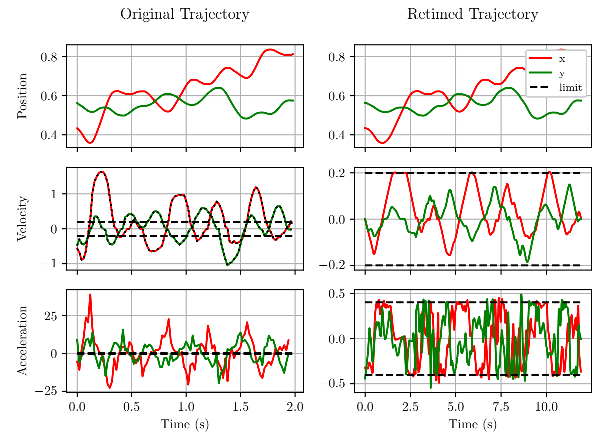

Fig. 7. Example of trajectory retiming. The original stroke trajectory (left) is retimed (right) to respect the velocity and acceleration limits (dashed-black). We apply time-optimal retiming, which minimizes the time taken to execute the trajectory subject to the constraints. Discretization causes minor violations.

of 0.2m/s and 0.4m/s^2, respectively, for each of x and y when the pen is down. Although we do not yet address it in this work, we believe joint state-action limits would be much more interesting, since there are many instances where the edges of the workspace do not permit as much control authority which we hope would be captured in the compositions.

The implementation details of the various approaches are as follows. For those using trajectory retiming (baseline, 1, 4), we apply the TOPP-RA algorithm [19, 20] with the velocity and acceleration limits (see Fig. 7). For approach 2 (diffusion in the control space), we rescale the network outputs using a sigmoid function to generate acceleration outputs in the range $[-0.4, 0.4]$ and integrate them to get velocities and positions (clipping velocity as needed). For approach 3, we use a simple relu barrier function to penalize each velocity and acceleration exceeding the limits.

C. Simulating Robot Execution

Fig. 1 simulates the robot execution using a velocity and force-limited feed-forward PID controller and a Forward Euler simulation with force control. The hard limits are set to 50% above the soft limits of $0.2m/s$ and $0.4m/s^2$.

IV. Discussion

We observe that different approaches will exhibit different performances for our two primary objectives: (1) dynamic feasibility and (2) artistic generation unique to the embodiment. The former is quantifiable and can be observed in Fig. 4 while the latter is more qualitative and can be observed in the sample trajectories from Fig. 3.

Subjectively, we observe Approaches 1 (training on retimed data) and 4 (classifier-free guidance) to perform the best in terms of balancing drawing quality and dynamic feasibility. Their similar performance is not surprising given that we have just 2 classes for classifier-free guidance. While the Baseline (apply retiming after generation) and Approach 2 (map output space to feasible region) both have less constraint violation (as expected, since they explicitly enforce the constraints), the baseline appears to produce less fluid paintings while Approach 2 produces overly simplistic ones. Based on previous experiments (omitted), we believe the task of generating lower-level control signals is more difficult to learn than generating higher-level state and action sequences. Finally, Approach 3 (classifier-guided diffusion) produces reasonable quality paintings but has the most constraint violation.

V. Conclusions and Future Work

In this work, we seek to generate robot trajectories for artistic painting which exploit the dynamics unique to a robot embodiment. We compare several approaches for incorporating robot dynamics into the artistic trajectory generation process and find that classifier-free guidance and training on dynamically feasible data both work well.

In the future, we plan to extend this work by incorporating more realistic dynamics constraints and executing the generated trajectories on the physical graffiti spray painting robot from [2, 3]. We also plan to generate more useable artistic trajectories through a number of architectural upgrades including conditioning on a target image, CLIP embedding, or class and replacing the 1D U-Net with an attention-based model for global context. Finally, we plan to more rigorously evaluate both the artistic qualities and dynamic feasibilities of the generated trajectories.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2008302.

References

[1] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sP1fo2K9DFG.

[2] Gerry Chen, Sereym Baek, Juan-Diego Florez, Wanli Qian, Sang-Won Leigh, Seth Hutchinson, and Frank Dellaert. GTGraffiti: Spray painting graffiti art from human painting motions with a cable driven parallel robot. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4065–4072, 2022. doi: 10.1109/ICRA46639.2022.9812008.

[3] Gerry Chen, Seth Hutchinson, and Frank Dellaert. Locally optimal estimation and control of cable driven parallel robots using time varying linear quadratic gaussian (lqg) control. https://gerry-chen.com/publications/Chen22iros_cdpr_control/Chen22iros_cdpr_tracking_control.pdf, 2022.

[4] Gerry Chen, Frank Dellaert, and Seth Hutchinson. Generalizing trajectory retiming to quadratic objective functions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[5] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[6] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press, 2005. ISBN 0-262-03327-5.

[7] Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. Béziersketch: A generative model for scalable vector sketches, 2020.

[8] Songwei Ge, Vedanuj Goswami, C. Lawrence Zitnick, and Devi Parikh. Creative sketch generation, 2020.

[9] Asya Grechka, Guillaume Couairon, and Matthieu Cord. GradPaint: Gradient-guided inpainting with diffusion models. *Computer Vision and Image Understanding*, 240:103928, 2024. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2024.103928. URL https://www.sciencedirect.com/science/article/pii/S1077314224000092.

[10] David Ha and Douglas Eck. A neural representation of sketch drawings. In *ICLR 2018*, 2018. 2018.

[11] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 1998.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[13] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

[14] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph.*, 39(6), nov 2020. ISSN 0730-0301. doi: 10.1145/3414685.3417871. URL https://doi.org/10.1145/3414685.3417871.

[15] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11451–11461, 2022. doi: 10.1109/CVPR52688.2022.01117.

[16] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[17] OpenAI. GPT-4V(ision) system card. Technical report, OpenAI, September 2023. URL https://cdn.openai.com/papers/GPTV_System_Card.pdf.

[18] OpenAI. Gpt-4o: Openai's multimodal model. https://openai.com/index/hello-gpt-4o/, 2024. Accessed: 2024-05-29.

[19] Hung Pham and Quang-Cuong Pham. A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis. *IEEE Transactions on Robotics*, 34(3):645–659, June 2018. ISSN 1941-0468. doi: 10.1109/TRO.2018.2819195.

[20] Hung Pham, Quang-Cuong Pham, Joseph Mirabel, and EdsterG. Toppra: Time-Optimal Path Parameterization, 2023. URL https://github.com/hungpham2511/toppra.

[21] Zhiyu Qu, Lan Yang, Honggang Zhang, Tao Xiang, Kaiyue Pang, and Yi-Zhe Song. Wired perspectives: Multi-view wire art embraces generative ai. In *CVPR*, 2024.

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

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[24] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. StyleCLIPDraw: Coupling content and style in text-to-drawing translation, 2022.

[25] Peter Schaldenbrand, James McCann, and Jean Oh. FRIDA: A collaborative robot painter with a differentiable, Real2Sim2Real planning environment. In *2023 IEEE International Conference on Robotics and Automa-*

tion (ICRA), pages 11712–11718, 2023. doi: 10.1109/ICRA48891.2023.10160702.

[26] Peter Schaldenbrand, Gaurav Parmar, Jun-Yan Zhu, James McCann, and Jean Oh. CoFRIDA: Self-supervised fine-tuning for human-robot co-painting. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[27] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nJfylDvgzlq.

[28] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. CLIPasso: Semantically-aware object sketching. *ACM Trans. Graph.*, 41(4), jul 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530068. URL https://doi.org/10.1145/3528223.3530068.

[29] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. CLIPascene: Scene sketching with different types and levels of abstraction, 2023.

[30] Qiang Wang, Haoge Deng, Yonggang Qi, Da Li, and Yi-Zhe Song. Sketchknitter: Vectorized sketch generation with diffusion models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4eJ43EN2g6l.