Closed-Loop Control of Direct Ink Writing via Reinforcement Learning

MICHAL PIOVARČI, IST Austria
MICHAEL FOSHEY, MIT CSAIL
JIE XU, MIT CSAIL
TIMOTHY ERPS, MIT CSAIL
VAHID BABAEI, MPI Informatics
PIOTR DIDYK, USI Lugano
SZYMON RUSINKIEWICZ, Princeton University
WOJCIECH MATUSIK, MIT CSAIL
BERND BICKEL, IST Austria

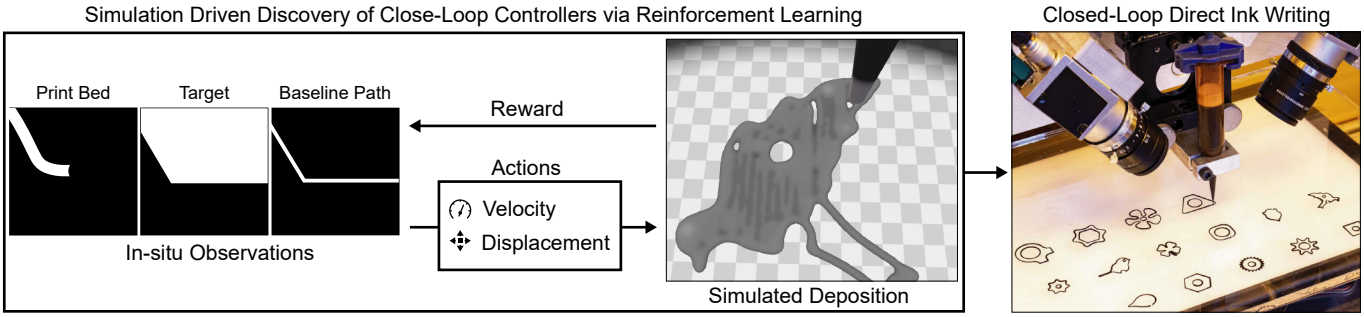


Fig. 1. We propose a numerical environment suitable for learning close-loop control strategies for additive manufacturing via direct ink writing. Our method observes an in-situ view of the printing process and adjusts the velocity and printing path to achieve the desired deposition. The control policies learned exclusively in simulation can be deployed on real hardware.

Enabling additive manufacturing to employ a wide range of novel, functional materials can be a major boost to this technology. However, making such materials printable requires painstaking trial-and-error by an expert operator, as they typically tend to exhibit peculiar rheological or hysteresis properties. Even in the case of successfully finding the process parameters, there is no guarantee of print-to-print consistency due to material differences between batches. These challenges make closed-loop feedback an attractive option where the process parameters are adjusted on-the-fly. There are several challenges for designing an efficient controller: the deposition parameters are complex and highly coupled, artifacts occur after long time horizons, simulating the deposition is computationally costly, and learning on hardware is intractable. In this work, we demonstrate the feasibility of learning a closed-loop control policy for additive manufacturing using reinforcement learning. We show that approximate, but efficient, numerical simulation is sufficient as long as it allows learning the behavioral patterns of deposition that translate to real-world experiences. In combination with reinforcement learning, our model can be used to discover control policies that outperform baseline controllers. Furthermore, the recovered policies have a minimal sim-to-real gap. We showcase this by applying our control policy in-vivo on a single-layer, direct ink writing printer.

1 INTRODUCTION

A critical component of manufacturing is identifying process parameters that consistently produce high-quality structures. In commercial devices, this is typically achieved by expensive trial-and-error experimentation [Gao et al. 2015]. To make such an optimization feasible, a critical assumption is made: the relationship between process parameters and printing outcome is predictable. However, such an assumption does not hold in practice because all manufacturing processes are stochastic in nature. Specifically, in additive manufacturing, variability in both materials and intrinsic process parameters can cause geometric errors leading to imprecision that can compromise the functional properties of the final prints. Therefore, transition to closed-loop control is indispensable for industrial adoption of additive manufacturing [Wang et al. 2020].

Recently, we have seen promising progress in learning policies for interaction with amorphous materials [Li et al. 2019b; Zhang et al. 2020]. Unfortunately, in the context of additive manufacturing, discovering effective control strategies is significantly more challenging. The deposition parameters have a non-linear coupling to the dynamic material properties. To assess the severity of deposition errors, we need to observe the material over long time horizons. Available simulators either lack predictive power [Mozaffar et al. 2018] or have prohibitive computational complexity for learning [Tang et al. 2018; Yan et al. 2018]. Moreover, learning on hardware is intractable as we require tens of thousands of printed samples. These challenges are further exaggerated by the limited perception of printing hardware, where typically, only a small in-situ view is available to assess the deposition quality.

In this work, we propose the first closed-loop controller for additive manufacturing trained purely in simulation that can be later deployed in real hardware. To achieve this we formulate a custom numerical model of the deposition process. Motivated by the limited

hardware perception we make a key assumption: a numerical model is sufficiently accurate as long as we can learn behavioral patterns that hold across both simulated and real environments. This allows us to replace physically accurate but prohibitively slow simulations with efficient approximations. To ameliorate the sim-to-real gap, we enhance the simulation with a data-driven noise distribution on the spread of the deposited material. We further show that careful selection of input and action space is necessary for hardware transfer. Lastly, we leverage the privileged information about the deposition process to formulate a reward function that encourages policies that account for material changes over long horizons. Thanks to the above advancements, our control policy can be trained exclusively in simulation with a minimal sim-to-real gap. We showcase this by deploying our policy on a custom single-layer direct ink writing printer. Finally, we demonstrate that our policy outperforms baseline deposition methods in simulation and physical hardware with low or high viscosity materials.

2 RELATED WORK

To identify process parameters for additive manufacturing, it is important to understand the complex interaction between a material and a deposition process. This is typically done through trial-and-error experimentation either on physical hardware [Baturynska et al. 2018; Kappes et al. 2018; Wang et al. 2018] or in simulation [Ogoke and Farimani 2021]. Recently, optimal experiment design and, more specifically, Gaussian processes have become a tool for efficient use of the samples to understand the deposition problem [Erps et al. 2021]. However, even though Gaussian Processes model the deposition variance, they do not offer tools to adjust the deposition on-the-fly. Another approach to improve the printing process is to design closed-loop controllers. One of the first designs was proposed by Sitthi-Amorn et al. [2015] that monitors each layer deposited by a printing process to compute an adjustment layer. Liu et al. [2017] built upon the idea and trained a discriminator that can identify the type and magnitude of observed defects. A similar approach was proposed by Yao et al. [2018] that uses handcrafted features to identify when a print significantly drops in quality. The main disadvantage of these methods is that they rely on collecting the in-situ observations to propose one corrective step by adjusting the process parameters. However, this means that the prints continue with sub-optimal parameters, and it can take several layers to adjust the deposition. In contrast, our system runs in-process and reacts to the in-situ views immediately. This ensures high-quality deposition and adaptability to material changes.

Recently machine learning techniques sparked a new interest in the design of adaptive control policies [Mnih et al. 2015]. A particularly successful approach for high-quality in-process control is to adopt the Model Predictive Control paradigm (MPC) [Gu et al. 2016; Nagabandi et al. 2018; Oh et al. 2017; Silver et al. 2017; Srinivas et al. 2018]. The control scheme of MPC relies on an observation of the current state and a short-horizon prediction of the future states. By manipulating the process parameters, we observe the changes in future predictions and can pick a future with desirable characteristics. Particularly useful is to utilize deep models to generate differentiable predictors that provide derivatives with respect to control changes [Li et al. 2019a; Schenck and Fox 2018; Toussaint et al. 2018]. However, addressing the uncertainties of the deposition process with MPC is challenging. In a noisy environment, we can rely only on the expected prediction of the deposition. This leads to a conservative control policy that effectively executes the mean action. Moreover, reacting to material changes over time requires optimizing actions for long time horizons which is a known weakness of the MPC paradigm [Garcia et al. 1989]. As a result, MPC is not suitable for in-process control in noisy environments.

Another option to derive control policies is to leverage deep reinforced learning [Akkaya et al. 2019; Lee et al. 2019; Liu and Hodgins 2018; Peng et al. 2018; Rajeswaran et al. 2017; Yu et al. 2019]. The key challenge in the design of such controllers is formulating an efficient numerical model that captures the governing physical phenomena. As a consequence, it is most commonly applied to rigid body dynamics and rigid robots where such models are readily available [Bender et al. 2014; Coumans and Bai 2016; Lee et al. 2018; Todorov et al. 2012; Xu et al. 2019]. In contrast, learning with non-rigid objects is significantly more challenging as the computation time for deformable materials is higher and relies on some prior knowledge of the task [Clegg et al. 2018; Elliott and Cakmak 2018; Ma et al. 2018; Wu et al. 2019]. Recently Zhang et al. [2020] proposed a numerical model for training control policies where a rigid object interacts with amorphous materials. Similarly, in our work a rigid printing nozzle interacts with the fluid-like printing material. However, our model is specialized for the printing hardware and models not only the deposition but also its variance. We demonstrate that this is an important component in minimizing the sim-to-real gap and design control policies that are readily applicable to the physical hardware.

3 HARDWARE PRELIMINARIES

The choice of additive manufacturing technology constraints the subsequent numerical modeling. To keep the applicability of our developed system as wide as possible, we opted for a direct write needle deposition system mounted on a 3-axis Cartesian robot (Figure 2 left). The robot allows us to freely control the acceleration and position of the dispenser. The dispenser can process a wide range of viscous materials, and the deposition is very similar to fused deposition modeling. We further enhance the apparatus with two camera modules. The cameras lie on the opposite sides of the nozzle to allow our apparatus to perceive the location around the deposition.

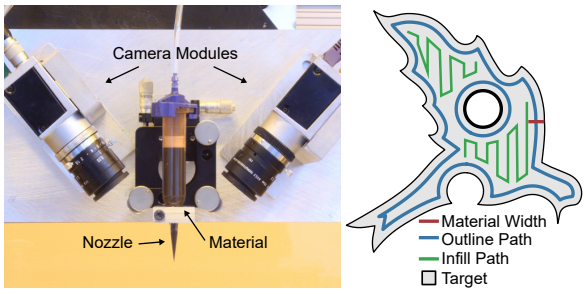


Fig. 2. The printing apparatus (left) and the baseline printing policy (right).

3.1 Baseline Controller

To control the printing apparatus, we employ a baseline slicer. The input to the slicer is a three-dimensional object. The output is a series of locations the printing head visits to reproduce the model as closely as possible. To generate a single slice of the object, we start by intersecting the 3D model with a Z-axis aligned plane (please note that this does not affect the generalizability since the input can be arbitrarily rotated). The slice is represented by a polygon that marks the outline of the printout (Figure 2 gray). To generate the printing path, we assume a constant width of deposition (Figure 2 red) that acts as a convolution on the printing path. The printing path (Figure 2 blue) is created by offsetting the print boundary by half the width of the material using the Clipper algorithm [Johnson 2015]. The infill pattern is generated by tracing a zig-zag line through the area of the print (Figure 2 green).

4 REINFORCEMENT LEARNING FOR ADDITIVE MANUFACTURING

The baseline control strictly relies on a constant width of the material. To discover policies that can adapt the baseline printing path to the in-situ observations, we formulate the search in a reinforcement learning framework. The control problem is described by a Markov decision process $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is the observation space, $\mathcal{A}$ is a continuous action space, $\mathcal{P} = P(s'|s, a)$ is the transition function that maps state s and action a to a new state s', and $\mathcal{R}(s, a) \rightarrow \mathbb{R}$ is the reward function. In the following section, we will describe how to apply this framework in the context of additive manufacturing.

4.1 Observation Space

The choice of observation space is critical for transferring the learned knowledge from simulation to physical hardware. A natural choice would be to utilize a direct image feed from a camera module. However, the large variety of available materials would introduce significant difficulty in the learning process where materials with similar physical behavior could be treated differently based on their appearance. Moreover, the rendering would need sufficient graphical fidelity to minimize the sim-to-real gap, further limiting learning efficiency. We propose to tackle these challenges by employing an engineered observation space. Rather than using the direct appearance feed from a camera module we process the signal into a heightmap. A heightmap is a 2D image where each pixel stores the height of the deposited material. For each height map location, the height is measured as a distance from the building plate to the deposited material. This allows our system to generalize to a wide range of sensors and materials.

We model our observation space as a small in-situ view centered at the printing nozzle. The view has a size of 3.5×3.5 mm. Since the location directly under the nozzle is obscured by physical hardware, we mask a small central position equivalent to $\frac{1}{7}$th of the view. Together with the local view we also supply the printer with the target and the baseline printing path in local view. Increasing the data efficiency, we make the observation that the images are rotation invariant along the direction of the printer's scheduled moving path. Therefore, we align the print direction with image X-axis. These inputs are stacked together into a 3-channel image, (Figure 3). For outline printing we threshold the heightfield to encourage a tighter fit to the target printout.

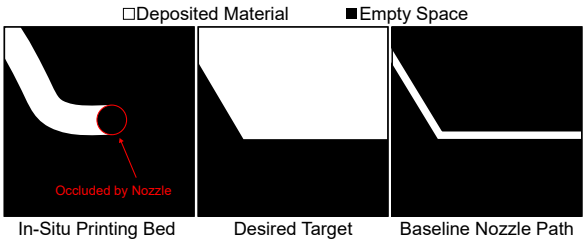
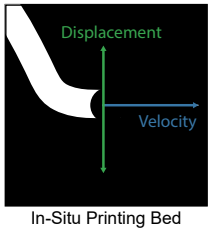


Fig. 3. Input to our policy are in-situ images of printing bed, target printout, and scheduled baseline printing path.

4.2 Action Space

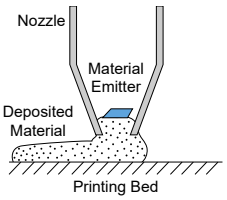
The selection of action space plays a critical role in applying the learned strategies on physical hardware. Tying the control scheme too closely to a physical setup would exaggerate the discrepancies between physical world and our numerical model. Moreover, the learned control strategies would be valid only for a particular hardware implementation and would not transfer across manufacturers of similar deposition systems. To address these issues we propose to learn high-level control strategies. More specifically, our policy tunes the velocity of the printing head and an offset from baseline printing path (see inset). Such control strategy allows us to decouple high-level goals from low-level inputs. Moreover, we can lift the hardware constraints to only require an apparatus with similar capabilities. In our simulation and physical samples we consider a velocity range of $[0.2, 2]$ mm/s and the displacement of ± 0.315 mm.



4.3 Transition Function

In our setting the transition function should approximate the deposition process. Unfortunately, this is a notoriously difficult problem that leads to prohibitive simulation complexities. To address this challenge we make a key assumption: a qualitative approximation of the deposition is sufficient as long as we can learn behavior patterns that translate to real-world experiences. To achieve this goal we propose to use an efficient numerical model. We enhance the model with a data-driven term that approximates the stochastic nature of the physical deposition. Such a combination allows us to efficiently discover control strategies that can adapt to deposition noise similar to the one observed in physical hardware.

To model the interaction of the deposited material with the printing apparatus we rely on Position-Based Dynamics (PBD) [Müller et al. 2007]. We model the printing materials a set of particles where each particle is defined by its position $\mathbf{p}$, velocity $\mathbf{v}$, mass m, and a set of constraints C. In our setting we consider two constraints: (1) collision with the



nozzle and (2) incompressibility of the fluid material. We model the collision with the nozzle as a hard inequality constraint:

$$C_i(\mathbf{p_i}) \coloneqq (\mathbf{p_i} - \mathbf{q}_c) \cdot \mathbf{n}_c \geq 0, \tag{1}$$

where $\mathbf{q}_c$ is the contact point of a particle with the nozzle geometry along the direction of particles motion $\mathbf{v}$ and $\mathbf{n}_c$ is the normal at the contact location. To ensure that our fluids remain incompressilbe we follow [Macklin and Müller 2013] and formulate a density constraint for each particle:

$$C_i(\mathbf{p}_1, ..., \mathbf{p}_n) \coloneqq \frac{\rho_i}{\rho_0} - 1 = 0, \tag{2}$$

$$\rho_i = \sum_j m_j W(\mathbf{p}_i - \mathbf{p}_j, h), \tag{3}$$

where ρ_0 is the rest density and ρ_i is given by a Smoothed Particle Hydrodynamics estimator [Müller et al. 2003] in which W is the smoothing kernel defined by the smoothing scale h.

Our idealized simulation deposits material at a constant rate. However, a physical hardware is not capable of such consistency. The dynamic material properties coupled with process errors introduce noise in the deposition process. To discover control strategies applicable to physical hardware we propose to reintroduce this noise into our numerical model. Due to the complex nature of the interactions between the material and the deposition process we propose to model the printing noise in a data-driven fashion. To formulate a predictive generative model we employ a tool from speech processing called Linear Predictive Coding (LPC) [Marple 1980]. Assuming a dynamic material flow rate Q modelled as a time-varying function we can predict the flow at time N as a weighted sum of M past flow samples and a noise term:

$$Q_N = -\sum_{m=1}^{M} a_{M,m} Q_{N-m} + \epsilon_n, \tag{4}$$

where Q_N are the flow samples, ϵ is the noise term, and $a_{M,m}$ are the parameters of M-th order auto-correlation filter. To find these coefficients Burg [1975] propose to minimize the following energies:

$$e_M = \sum_{k=1}^{N-m} |f_{M,k}|^2 + \sum_{k=1}^{N-m} |b_{M,k}|^2, \tag{5}$$

$$f_{M,k} = \sum_{i=0}^{M} a_{M,i} Q_{k+M-i}, \tag{6}$$

$$b_{M,k} = \sum_{i=0}^{M} a^*_{M,i} Q_{k+i}, \tag{7}$$

where $*$ denotes the complex conjugate. After finding the filter coefficients with Equation 5 we can synthesize new width variations with similar frequency composition to the physical hardware by filtering a buzz modeled as a white Gaussian noise. To fit the model we repeatedly printed an exemplar slice (Figure 4 left) and measured its width at fixed intervals (Figure 4 middle). Since we sample at discrete intervals we further smooth our data with an interpolating curve (Figure 4 right).

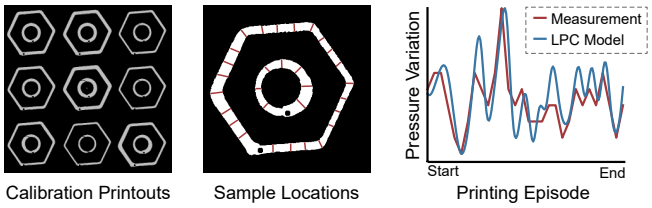


Fig. 4. We performed nine printouts and measured the width variation at specified locations. We fit the measured data with an LPC model. Please note that since our model is generative, we do not exactly match the data any observed resemblance is a testament to the quality of our predictor.

4.4 Reward Function

Viscous materials take significant time to settle after deposition. Therefore, to assess deposition errors, it is needed to observe the material spread over long horizons. However, the localized nature of the in-situ view does not permit such observations. At print-time, the observation window is limited by the velocity to the range of 0.875 to 8.75 seconds. In contrast, a low-viscosity material typically settles within 15 seconds of deposition. As a result, evaluating the deposition on the physical hardware is feasible only with a scan at the end of the print. In Section 5.3 we will show that such a constraint on performance calculation inhibits the discovery of optimal control strategies. To learn effective control policies over long-horizons we propose to leverage the privileged information available only in the simulated environment. At each timestep, we calculate the reward as a printing performance measured on the entire printing bed. More specifically, given an image of the printing bed C and desired target printout $\mathcal{T}$ we estimate the reward $\mathcal{R}^t$ at simulation step t:

$$\mathcal{R}^t = \sum C\mathcal{T} - \sum C(1 - \mathcal{T}), \tag{8}$$

where the first term rewards depositing material within the target area and the second punishes over-deposition. For infill printing we add an additional reward term to encourage deposition with minimal height variation:

$$\mathcal{R}^t = \sum C\mathcal{T} - \sum C(1 - \mathcal{T}) - std(C\mathcal{T}). \tag{9}$$

To further accelerate the learning we generate dense rewards as a delta between two steps $\mathcal{R} = \mathcal{R}^{t+1} - \mathcal{R}^t$.

5 RESULTS

In this section, we provide results obtained in both virtual and physical environments. We first show that an adaptive policy can outperform baseline approaches in environments with constant deposition. Next, we showcase the in-process monitoring and the ability of our policy to adapt to dynamic environments. Finally, we demonstrate our learned controllers transferring to the physical world with a minimal sim-to-real gap.

5.1 Comparison With Baseline Controller

We evaluate the optimized control scheme on a selection of freeform and CAD models sampled from Thingy10k [Zhou and Jacobson 2016] and ABC [Koch et al. 2019] datasets (see Appendix A). In total, we have 113 unseen slices corresponding to 96 unseen geometries.

We report our findings in Figure 5. For each input slice, we report improvement on the printed boundary as the average offset. The average offset is defined as a sum of areas of under and over deposited material normalized by the outline length. More specifically, given an image of the target slice $\mathcal{T}$, printed canvas C, and the length of the outline l, the average offset O is computed as:

$$O = \frac{\sum(1-C)\mathcal{T}}{l} + \frac{\sum C(1-\mathcal{T})}{l}. \tag{10}$$

The improvement is calculated as a difference between the baseline and our policy. Therefore, a value higher than zero indicates that our control policy outperformed the baseline. As we can see, our policy achieved better performance in all considered models.

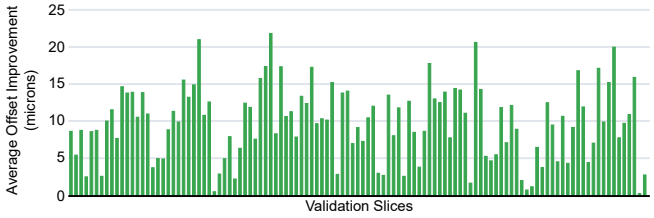


Fig. 5. The relative improvement of our policy over baseline in printing task with constant deposition.

5.2 Performance in Dynamic Environments

We evaluate our controller in environments with stochastic material flow. To perform quantitative evaluation we utilize a single flow variation profile. We use the same evaluation dataset as for constant-flow policy and report the overall improvement over the baseline controller, (Figure 6). We can observe that in each of the considered slices, our closed-loop controller outperformed the baseline.

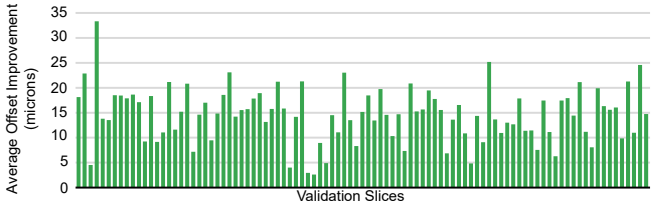


Fig. 6. The relative improvement of our policy over baseline in printing task with noisy deposition.

5.2.1 Outline Printing. To evaluate the quality of our printous we analyze the overflow and underflow histograms on a subset of the evaluation dataset, (Figure 7). We can observe that the deposition histograms of our policy are narrower than the baseline. The average improvement in the standard deviation was measured at 16 microns. Moreover, the deposition histograms generated by our policy more closely approximate a normal distribution with an average skewness 0.5 lower that that of the baseline. We can therefore conclude that our control policy achieves a tighter control on the material deposition.

5.2.2 Infill Printing. We have also evaluated the infill policy in a noisy environment, (Figure 8). We can observe that the deposition noise leads to an accumulation of material. The accumulation

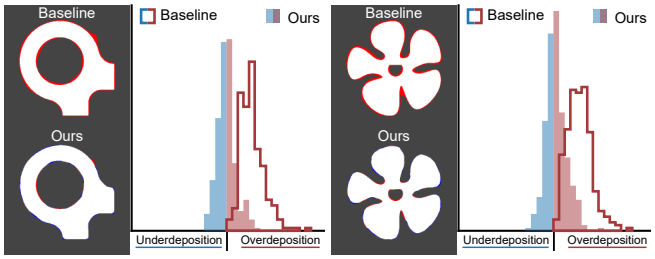


Fig. 7. Deposition histograms for two exemplar slices from our dataset. Our control policy achieves tighter control over the deposition process even in challenging noisy environments.

eventually results in a bulge of material in the center of the print, complicating the deposition of subsequent layers as the material would tend to slide off. In contrast, our policy dynamically adjusts the printing path to generate a print with significantly better height uniformity. As we can observe, the surface generated by our policy is almost flat and ready for deposition of additional layers.

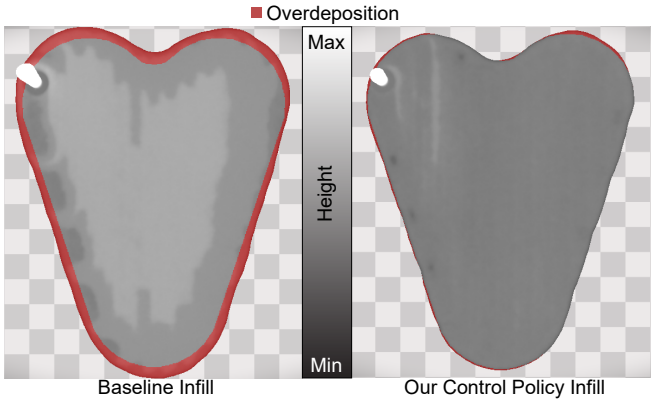


Fig. 8. In a noisy environment the baseline printing policy (left) significantly over-deposits and does not produce a uniform surface. In contrast, our policy (right) has almost no over-deposition and creates an almost uniform surface suitable for multi-layer printing.

5.3 Ablation Studies

To demonstrate that our design decision indeed lead to improvements to learning the deposition we conduct several ablation studies that test the individual components: observation space, action space, and reward function. Finally, we investigate how our learned controllers generalize across materials with varying viscosity.

5.3.1 Ablation Study on Observation Space. Our control policy relies on a live view of the deposition system to select the control parameters. However, the in-situ view is a technologically challenging addition to the printer hardware that requires a carefully calibrated imagining setup. With this ablation study, we verify how important the individual observations are to the final print quality. We consider three cases: (1) no printing bed view, (2) no target view, and (3) no future path view. We analyzed the results from the pre-test (full observation space $\mu = 9.7$, $\sigma = 4.9$) and the post-tests (no canvas $\mu = 8.8$, $\sigma = 5.7$, no target $\mu = 7.2$, $\sigma = 5.5$, no path $\mu = 8.4$,

$\sigma = 4.8$) printing task using paired t-tests with Holm-Bonferroni correction. The analysis indicates that the availability of all three inputs: the printing bed, the target, and the path improved final printouts ($P < 0.01$ for all three cases).

5.3.2 Ablation Study on Action Space. To evaluate the need to tweak both the printing velocity and the printing path, we trained two control policies with a limited action set to either alter the velocity or path offset. We analyzed the results from the pre-test (full action space $\mu = 12.7$, $\sigma = 5.7$) and the post-tests (velocity $\mu = 7.5$, $\sigma = 2.5$, displacement $\mu = 5.6$, $\sigma = 8.3$) printing task using paired t-tests with Holm-Bonferroni correction. The analysis indicates that the availability of the full action space resulted in an improvement in final printouts ($P < 0.001$ for both cases).

The difference in performance depends on the inherent limitations of the individual actions. On the one hand, adjusting velocity is fast (under 6.6 milliseconds) but can cope only with moderate changes in material width. This can be observed as the larger bulges of over-deposited material, (Figure 9 dark green region). On the other hand, while offset can cope with larger material differences it needs between 0.13 and 1.3 seconds to adjust. As a result, offset adjustment cannot cope with sudden material changes, (Figure 9 light green region). In contrast, by utilizing the full action space our policy can combine the advantages of the individual actions and minimize over-deposition.

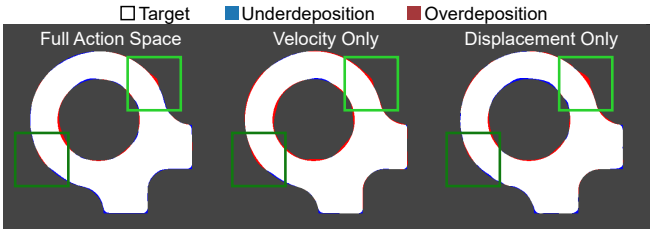


Fig. 9. Ablation of action space shows that fast velocity updates are advantageous in regions with fast material changes (light green) whereas displacement updates can cope with large material deviations (dark green). By utilizing both actions our control policy can effectively handle both scenarios.

5.3.3 Ablation Study on Reward Function. Our reward function uses privileged information from the numerical simulation to evaluate how material settles over time. However, such information is not readily available on physical hardware. One either evaluates the reward once at the end of each episode to include material flow or at each timestep by disregarding long-term material motion. We evaluated how such changes to the reward function would affect our control policies. We analyzed the results from the pre-test (privileged reward $\mu = 12.7$, $\sigma = 5.7$) and the post-tests (delayed reward $\mu = -22.3$, $\sigma = 8.6$, immediate reward $\mu = 9.2$, $\sigma = 8.0$) printing task using paired t-tests with Holm-Bonferroni correction. The analysis indicates that the availability of the privileged information resulted in an improvement in final patterns ($P < 0.001$ for both cases).

The learning process for a delayed reward is significantly slower, and it is unclear if performance similar to our policy can be achieved. On the other hand, the immediate reward policy learns faster but cannot handle material changes over longer time horizons, (Figure 10).

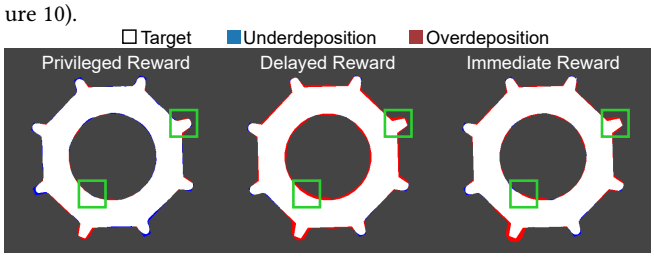


Fig. 10. Ablation of reward function demonstrates that the privileges information accelerates learning and allows the control policy to adapt to long-horizon material changes resulting in better printouts in challenging regions (green).

5.3.4 Ablation Study on Viscosity. To verify that our policy can adapt to various materials, we trained three models of varying viscosity, (Figure 11). We can observe that, without an adaptive control scheme, changing the material causes local over- or under-deposition. Our trained policy dynamically adjusts the offset and velocity to counterbalance the changes in the deposition. We can see that our policy is particularly good at handling smooth width changes and quickly recovers from a spike in printing width.

We further observe how our controllers handle deviations from the training material. The policy learned on low-viscosity materials consistently under-deposits when used to print at higher viscosities. Conversely, the control policy learned on high-viscosity material over-deposits when applied to materials with lower viscosities. From this observation we conclude that our policy learns the spread of the material post-deposition and uses this information to guide the printing process. Therefore, small viscosity variations are not likely to pose significant challenge for our learned policies. However, if the learned material behavior is significantly violated the in-situ observation space limits the ability of our policy to adapt to a before unseen material.

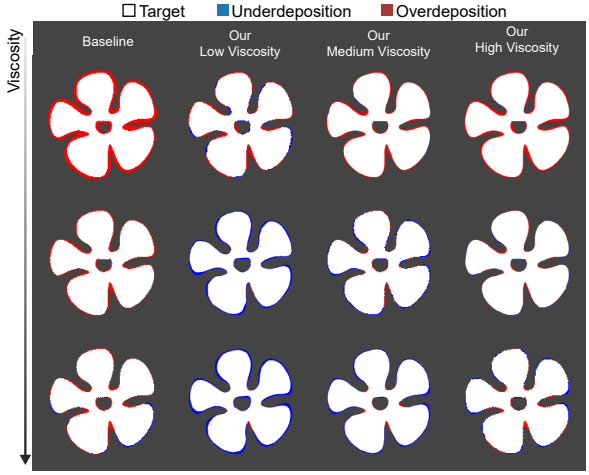


Fig. 11. We trained our policy with low, medium, and a high viscosity material. We show how the individual trained policies react to changes in material properties.

5.4 Performance on Physical Hardware

Finally, we evaluate our control policies on physical hardware. The policies were trained exclusively in simulation without any additional fine-tuning on the printing device. To conduct the evaluation, we equipped our printer with a pressure controller. The pressure control was set to a sinusoidal oscillatory signal to provide a controllable dynamic change in material properties. We used two materials, with high and low viscosity, and used two separate policies pretrained in simulation using those materials. We printed 22 slices, of which 11 corresponded to the simulation training set and 11 to the evaluation set. We monitor the printing process and use the captured images to run our evaluation function to capture quantitative results. We observe that our controllers improve the average offset over the baseline print in every scenario, (Figure 12).

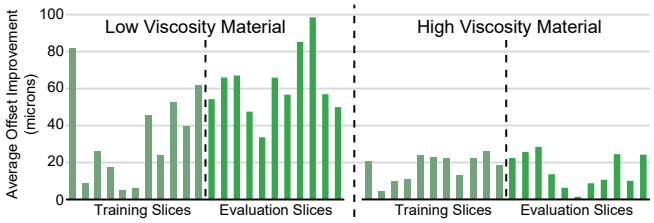


Fig. 12. The relative improvement of our policy over baseline in physical printing task.

A sample of the fabricated slices can be seen in Figure 13. The print target (white) is overlayed with a map of underdeposited (blue) and overdeposited (red) material. We further plot a histogram of under and over deposition.

We can see that our control policy transferred remarkably well to the physical hardware without any additional training. Our policy consistently achieves smaller over-deposition while not suffering from significant under-deposition. Moreover, in many cases our policy achieves histograms with smaller width suggesting we achieved a tighter control over the material deposition than the baseline. This demonstrates that our numerical model enables learning control policies for additive manufacturing in simulation.

6 CONCLUSION

We present a methodology for learning closed-loop control strategies for direct ink writing via reinforcement learning. To learn an effective control policy, we propose a custom numerical model of the deposition process. During the design of our model, we tackle several challenges. To obtain an efficient numerical simulator, we leverage the assumption that a numerical model is sufficiently accurate when it allows to learn behavioral patterns that translate to the physical task. To include non-linear coupling between process parameters and printed materials, we utilize a data-driven predictive model for the deposition imperfections. Finally, to enable long horizon learning with viscous materials, we use the privileged information generated by our numerical model for reward computation. In several ablation studies, we show that these components are required to achieve high-quality printing, effectively react to instantaneous and long horizon material changes, handle materials with varying viscosity, and adapt the deposition parameters to achieve printouts with minimal over-deposition and smooth top layers.

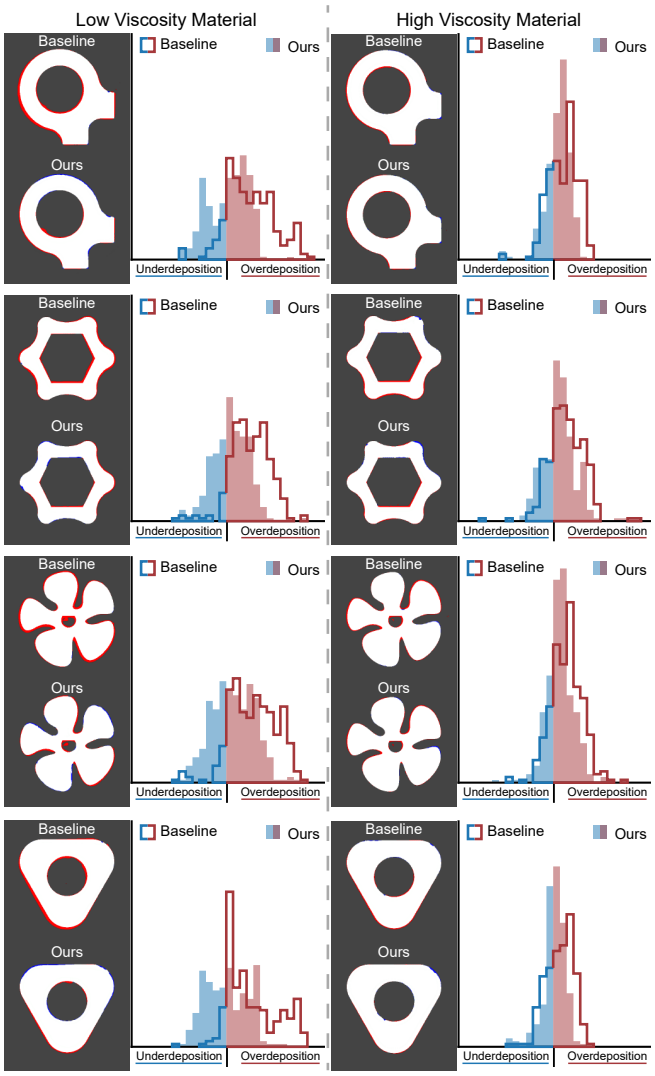


Fig. 13. Deposition quality estimation of physical result manufactured with baseline and our learned policy.

We demonstrate that our model can be used to train control policies that outperform baseline controllers, and transfer to physical apparatus with a minimal sim-to-real gap. We showcase this by applying control policies trained exclusively in simulation on a physical printing apparatus. We use our policies to fabricate several prototypes using low and high viscosity materials. The quantitative and qualitative analysis clearly shows the improvement of our controllers over baseline printing. This indicates that our numerical model can guide the future development of closed-loop policies for additive manufacturing. Thanks to its minimal sim-to-real gap, the model democratizes research on learning for additive manufacturing by limiting the need to invest in specialized hardware.

We optimized our control policies assuming single-layer deposition. Although this is sufficient for many applications, e.g., printed

electronics, microfluidics, or bio-printing, the clear next step is multi-layer, 3D printing. To achieve this, we are improving our acquisition setup for capturing taller, more complex geometries. This will open up a plethora of new research directions, such as slicing-aware path planning and applications, such as printing optical designs, food, or functional mechanisms. We believe our introduced approach can serve as a blueprint for the future research in AI-driven control of advanced manufacturing technologies, such as machining and laser-material processing. An important lesson we learned is the value of efficient numerical simulation of the involved physical phenomena. The graphics community can be a major driving force behind developing such simulations.

REFERENCES

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. 2019. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113* (2019).

Ivanna Baturynska, Oleksandr Semeniuta, and Kristian Martinsen. 2018. Optimization of process parameters for powder bed fusion additive manufacturing by combination of machine learning and finite element method: A conceptual framework. *Procedia Cirp* 67 (2018), 227–232.

Jan Bender, Matthias Müller, Miguel A Otaduy, Matthias Teschner, and Miles Macklin. 2014. A survey on position-based simulation methods in computer graphics. In *Computer graphics forum*, Vol. 33. Wiley Online Library, 228–251.

John Parker Burg. 1975. *Maximum Entropy Spectral Analysis.* Stanford University.

Alexander Clegg, Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. 2018. Learning to dress: Synthesizing human dressing motion via deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–10.

Erwin Coumans and Yunfei Bai. 2016. Pybullet, a python module for physics simulation for games, robotics and machine learning. (2016).

Sarah Elliott and Maya Cakmak. 2018. Robotic cleaning through dirt rearrangement planning with learned transition models. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1623–1630.

Timothy Erps, Michael Foshey, Mina Konaković Luković, Wan Shou, Hanns Hagen Goetzke, Herve Dietsch, Klaus Stoll, Bernhard von Vacano, and Wojciech Matusik. 2021. Accelerated Discovery of 3D Printing Materials Using Data-Driven Multi-Objective Optimization. arXiv:2106.15697 [cond-mat.mtrl-sci]

Wei Gao, Yunbo Zhang, Devarajan Ramanujan, Karthik Ramani, Yong Chen, Christopher B Williams, Charlie CL Wang, Yung C Shin, Song Zhang, and Pablo D Zavattieri. 2015. The status, challenges, and future of additive manufacturing in engineering. *Computer-Aided Design* 69 (2015), 65–89.

Carlos E Garcia, David M Prett, and Manfred Morari. 1989. Model predictive control: Theory and practice—A survey. *Automatica* 25, 3 (1989), 335–348.

Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. 2016. Continuous deep q-learning with model-based acceleration. In *International conference on machine learning*. PMLR, 2829–2838.

Angus Johnson. 2015. Clipper - an open source freeware library for clipping and offsetting lines and polygons. http://www.angusj.com/delphi/clipper.php.

Branden Kappes, Senthamilaruvi Moorthy, Dana Drake, Henry Geerlings, and Aaron Stebner. 2018. Machine learning to optimize additive manufacturing parameters for laser powder bed fusion of Inconel 718. In *Proceedings of the 9th International Symposium on Superalloy 718 & Derivatives: Energy, Aerospace, and Industrial Applications*. Springer, 595–610.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. 2019. ABC: A Big CAD Model Dataset For Geometric Deep Learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jeongseok Lee, Michael X Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S Srinivasa, Mike Stilman, and C Karen Liu. 2018. Dart: Dynamic animation and robotics toolkit. *Journal of Open Source Software* 3, 22 (2018), 500.

Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. 2019. Scalable muscle-actuated human simulation and control. *ACM Transactions On Graphics (TOG)* 38, 4 (2019), 1–13.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. 2019a. Learning Particle Dynamics for Manipulating Rigid Bodies, Deformable Objects, and Fluids. In *International Conference on Learning Representations*.

Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B Tenenbaum, Antonio Torralba, and Russ Tedrake. 2019b. Propagation networks for model-based control under partial observation. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 1205–1211.

Chenang Liu, David Roberson, and Zhenyu Kong. 2017. Textural analysis-based online closed-loop quality control for additive manufacturing processes. In *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE), 1127–1132.

Libin Liu and Jessica Hodgins. 2018. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Pingchuan Ma, Yunsheng Tian, Zherong Pan, Bo Ren, and Dinesh Manocha. 2018. Fluid directed rigid body control using deep reinforcement learning. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–11.

Miles Macklin and Matthias Müller. 2013. Position based fluids. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

Larry Marple. 1980. A new autoregressive spectrum analysis algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, 4 (1980), 441–454. https://doi.org/10.1109/TASSP.1980.1163429

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

Mojtaba Mozaffar, Arindam Paul, Reda Al-Bahrani, Sarah Wolff, Alok Choudhary, Ankit Agrawal, Kornel Ehmann, and Jian Cao. 2018. Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks. *Manufacturing letters* 18 (2018), 35–39.

Matthias Müller, David Charypar, and Markus H Gross. 2003. Particle-based fluid simulation for interactive applications.. In *Symposium on Computer animation*. 154–159.

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118.

Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. 2018. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 7559–7566.

Francis Ogoke and Amir Barati Farimani. 2021. Thermal control of laser powder bed fusion using deep reinforcement learning. *Additive Manufacturing* 46 (2021), 102033. https://doi.org/10.1016/j.addma.2021.102033

Junhyuk Oh, Satinder Singh, and Honglak Lee. 2017. Value Prediction Network. In *NIPS*.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. 2017. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087* (2017).

Connor Schenck and Dieter Fox. 2018. Spnets: Differentiable fluid dynamics for deep neural networks. In *Conference on Robot Learning*. PMLR, 317–335.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

David Silver, Hado Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, et al. 2017. The predictron: End-to-end learning and planning. In *International Conference on Machine Learning*. PMLR, 3191–3199.

Pitchaya Sitthi-Amorn, Javier E Ramos, Yuwang Wangy, Joyce Kwan, Justin Lan, Wenshou Wang, and Wojciech Matusik. 2015. MultiFab: a machine vision assisted platform for multi-material 3D printing. *Acm Transactions on Graphics (Tog)* 34, 4 (2015), 1–11.

Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. 2018. Universal planning networks: Learning generalizable representations for visuomotor control. In *International Conference on Machine Learning*. PMLR, 4732–4741.

Chao Tang, Jie Lun Tan, and Chee How Wong. 2018. A numerical investigation on the physical mechanisms of single track defects in selective laser melting. *International Journal of Heat and Mass Transfer* 126 (2018), 957–968.

Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5026–5033.

Marc A Toussaint, Kelsey Rebecca Allen, Kevin A Smith, and Joshua B Tenenbaum. 2018. Differentiable physics and stable modes for tool-use and manipulation planning. (2018).

Chengcheng Wang, Xipeng Tan, Erjia Liu, and Shu Beng Tor. 2018. Process parameter optimization and mechanical properties for additively manufactured stainless steel 316L parts by selective electron beam melting. *Materials & Design* 147 (2018), 157–166.

Chengcheng Wang, XP Tan, SB Tor, and CS Lim. 2020. Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing* (2020), 101538.

Yilin Wu, Wilson Yan, Thanard Kurutach, Lerrel Pinto, and Pieter Abbeel. 2019. Learning to manipulate deformable objects without demonstrations. *arXiv preprint*

arXiv:1910.13439 (2019).
Jie Xu, Tao Du, Michael Foshey, Beichen Li, Bo Zhu, Adriana Schulz, and Wojciech Matusik. 2019. Learning to fly: computational controller design for hybrid UAVs with reinforcement learning. *ACM Transactions on Graphics (TOG)* 38, 4 (2019).
Wentao Yan, Ya Qian, Wenjun Ge, Stephen Lin, Wing Kam Liu, Feng Lin, and Gregory J Wagner. 2018. Meso-scale modeling of multiple-layer fabrication process in selective electron beam melting: inter-layer/track voids formation. *Materials & Design* 141 (2018), 210–219.
Bing Yao, Farhad Imani, and Hui Yang. 2018. Markov decision process for image-guided additive manufacturing. *IEEE Robotics and Automation Letters* 3, 4 (2018), 2792–2798.
Ri Yu, Hwangpil Park, and Jehee Lee. 2019. Figure skating simulation from video. In *Computer graphics forum*, Vol. 38. Wiley Online Library, 225–234.
Yunbo Zhang, Wenhao Yu, C Karen Liu, Charlie Kemp, and Greg Turk. 2020. Learning to manipulate amorphous materials. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–11.
Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797* (2016).

A METHODS

In the appendix we describe our hardware apparatus and its calibration in Section A.1. In Section A.2 we continue by providing more details about the simulation environment and specific discretization choices to enable faster learning of printing policies. Next, in Section A.3, we describe the employed learning model, training datasets, and showcase the training convergence. We finish by showing results of outline printing in simulation (Section A.4), as well as on the physical apparatus, (Section A.5).

A.1 Hardware Setup

In this work, we developed a direct write 3D printing platform with an optical feedback system that can measure the dispensed material real-time, in-situ. The 3D printer is comprised of a pressure driven syringe pump and pressure controller, a 3-axis Cartesian robot, optical imaging system, back-lit build platform, 3D-printer controller and CPU Figure 14. The 3-axis Cartesian robot is used to locate the build platform in x and y-direction and the print carriage in the z-direction. The pressure driven syringe pump and pressure controller are used to dispense and optically opaque material onto the back-lit build platform. The back-lit platform is used to illuminate the dispensed material. The movement of the robot, actuation of the syringe pump and timing of the cameras are controlled via the controller. The CPU is used to process the images after they are acquired and compute updated commands to send to the controller.

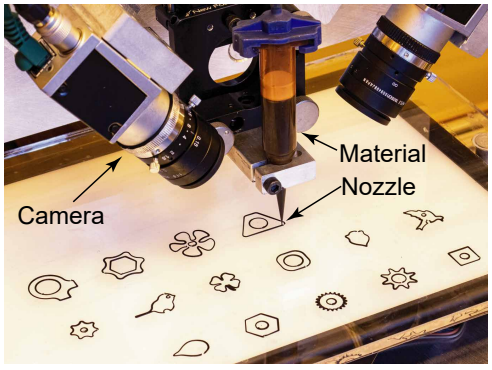


Fig. 14. The printing apparatus consisting of a 3-axis Cartesian robot, a direct write printing head, and a camera setup.

To enable realtime control of the printing process we implemented an in-situ view of the material deposition. Ideally we would capture a top-down view of the deposited material. Unfortunately, this is not possible since the material is obstructed by the dispensing nozzle. As a result the camera has to observe the printing bed from an angle. Since the nozzle would obstruct the view of any single camera we opted to use two cameras. More specifically, we place two CMOS cameras (Basler AG, Ahrensburg, Germany) at 45 degrees on each side of the dispensing nozzle, (Figure 14). We calibrate the camera by collecting a set of images and estimating its intrinsic parameters, (Figure 15 calibration). To obtain a single top-down view we capture a calibration target aligned with the image frames of both cameras, (Figure 15 homography). By calculating the homography between the captured targets and an ideal top-down view we can stitch the images into a single view from a virtual over-the-top camera. Finally, we mask the location of each nozzle in the image (Figure 15 nozzle masks) and obtain the final in-situ view, (Figure 15 stitched image).

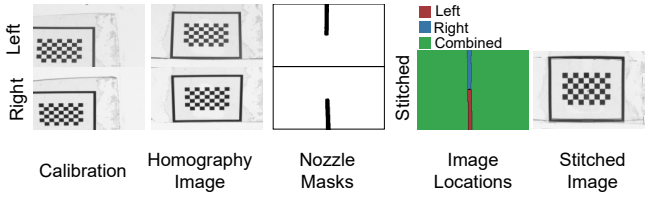


Fig. 15. The calibration of the imaging setup. First intrinsic parameters are estimated from calibration patterns. Next we compute the extrinsic calibration by calculating homographies between the cameras and an overhead view. We extract the masks by thresholding a photo of the nozzle. The final stitched image consists of 4 regions: (1) view only in left camera, (2) view only in right camera, (3) view in both cameras, (4) view in no camera. The final stitched image is show on the right.

The recovered in-situ view is scaled to attain the same universal scene unit size as our control policies are trained in. Since we seek to model the deposition only qualitatively it is sufficient to rescale the in-situ view to match the scale of the virtual environments. We identify this scaling factor separately for each material. To calibrate a single material we start by depositing a straight line at maximum velocity. The scaling factor is then the ratio required to match the observed thickness of the line with simulation. To extract the thickness of the deposited material we rely on its translucency properties. More precisely, we correlate material thickness with optical intensity, (Figure 16). We do this by depositing the material at various thickness and taking a picture with our camera setup. The optical intensity then decays exponentially with increased thickness which is captured by a power law fit.

The last assumption of our control policy is that the deposition needle is centered with respect to the in-situ view. To ensure that this assumption holds with the physical hardware we calibrate the location of the dispensing needle within the field of view of each camera and with respect to the build platform. First, a dial indicator is used to measure the height of the nozzle in z and the fine adjustment stage is adjusted until the nozzle is 254 microns above the print platform. Next, using a calibration target located on the build platform and the fine adjustment stage, the nozzle is centered in the field of view of each camera. This calibration

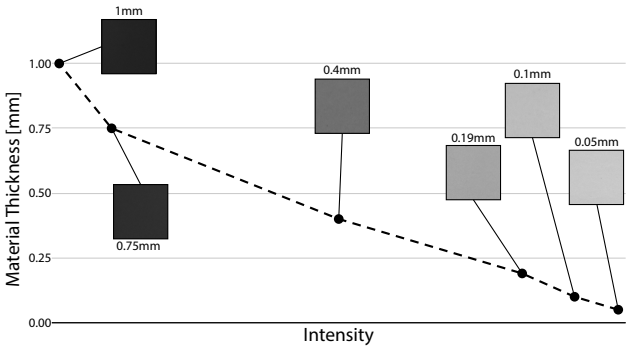


Fig. 16. Calibration images for correlating deposited material thickness with optical intensity and the corresponding fit.

procedure is done each time the nozzle is replaced during the start of each printing session.

A.2 Simulation Details

We replicate our printing apparatus in the simulation, (Figure 17). We model the nozzle as a collision object with a hard contact constraint on the fluid particles. Since modeling a pressurized reservoir is computationally costly as it requires us to have many particles in constant contact, we chose to approximate the deposition process at the peak of the nozzle. More specifically, we model the deposition as a particle emitter. To set the volume and velocity of the particles, we use a flow setting. The higher the flow, the more particles with higher initial velocities are generated. This qualitatively approximates the deposition process with a pressurized reservoir. The particle emitter is placed slightly inside the nozzle to allow for realistic material buildup and a delayed stop, similar to extrusion processes. Finally, we consider the printer to have only a finite acceleration per timestep. To accelerate to target velocity, we employ a linear acceleration scheme.

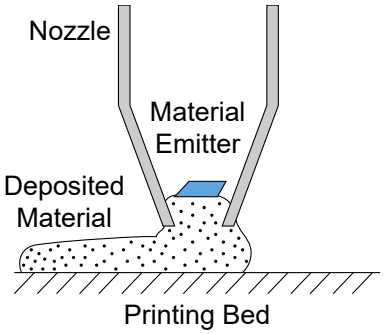


Fig. 17. Cross section of our simulation setup. The deposition system is modeled as a material emitter and a nozzle geometry. The deposited material accumulates both on the printing bed as well as inside the nozzle leading to delayed response similar to physical hardware.

Another important choice for the numerical model is the used discretization. We have two options: (1) time-based and (2) distance-based. We originally experimented with time-based discretization. However, we found out that time discretization is not suitable for printer modeling. As the velocity in simulation approaches zero, the difference in deposited material becomes progressively smaller until the gradient information completely vanishes, (Figure 18 left). Moreover, a time-based discretization allows the policy to affect the number of evaluations of the environment directly. As a result, it can avoid being punished for bad material deposition by quickly rushing the environment to finish. Considering these factors we opted for distance-based discretization, (Figure 18 right). The policy specifies the desired velocity at each interaction point, and the environment travels a predefined distance (0.315 mm) at the desired speed. This helps to regularize the reward function and enable learning of varying control policies.

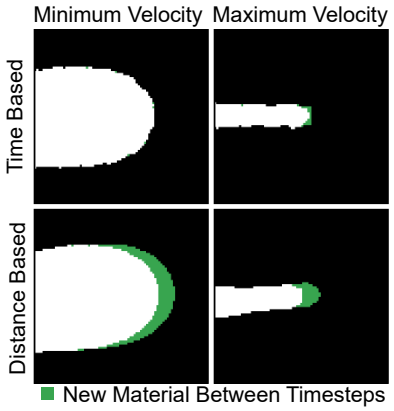


Fig. 18. The amount of material deposited between two simulation steps depends on the discretization choice. With time based discretization the reward gradient vanishes at low speeds. In contrast, distance based discretization produces a more uniform response.

An interesting design element is the orientation of the control polygons created by the slicer. When the outline is defined as points given counter-clockwise, then due to the applied rotation, each view is split roughly into two half-spaces, (Figure 19). The bottom one corresponds to outside i.e., generally black, and the upper one corresponds to inside i.e., generally white. However, the situation changes when outlining a hole. When printing a hole the two half-spaces swap location. We can remove this disambiguity by changing the orientation of the polylines defining holes in the model. By orienting them clockwise, we will effectively swap the two half-spaces to the same orientation as when printing the outer part. As a result, we achieve a better usage of trajectories and a more robust control scheme that does not need to be separately trained for each print's outer and inner parts.

A.3 Training Details

To train our control policy we start with g-code generated by a slicer. As inputs to the slicer we consider a set of 3D models collected from the Thingy10k dataset. To train a controller the input models need to be carefully selected. On the one hand, if we pick an object with too low frequency features with respect to the printing nozzle size

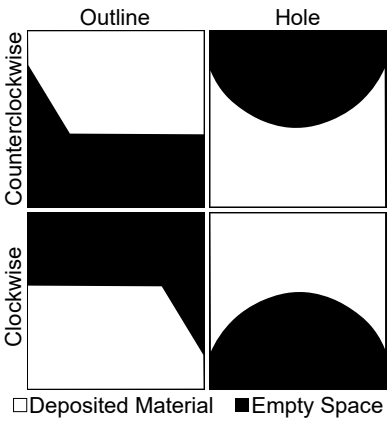


Fig. 19. The observation space is split into two half-spaces corresponding to target printout and void. To minimize the observation space and reuse trajectories it is therefore advantageous to pick different printing direction for objects outline and holes inside. By picking different directions we guarantee that material is always deposited only on one side.

then any printing errors due to control policy will have negligible influence on the final result. On the other hand, if we pick a model with too high frequency features with respect to the printing nozzle then the nozzle will be physically unable to reproduce these features. As a result we opted for a manual selection of 18 models that span a wide variety of features, (Figure 20). Each model is scaled to fit into a printing volume of 22×22 mm and sliced at random locations.

Fig. 20. Models contained in our training curriculum.

Our policy is represented as a CNN modeled after Mnih et al. [2015]. The network input is a $84 \times 84 \times 3$ image. The image is passed through three hidden layers. The convolution layers have the respective parameters: (32 filters, filter size 8, stride 4), (64 filters, filter size 4, stride 2), and (64 filters, filter size 3, stride 1). The final convolved image is linearized and passed through a fully-connected layer with 512 neurons that is connected to the output action. Each hidden layer uses the nonlinear rectifier activation. We formulate our objective function as:

$$\arg\max_{\theta} \mathbb{E}_t^C \left[\frac{\pi_{\theta_t}(a_t|s_t)}{\pi_{\theta_{t-1}}(a_t|s_t)} \hat{A}_t \right], \tag{11}$$

where t is a timestep in the optimization, θ are the hyperparameters of a neural network encoding our policy π that generates an action a_t based on a set of observations s_t, $\hat{A}_t$ is the estimator of the advantage function and the expectation $\mathbb{E}_t^C$ is an average of a finite batch of samples generated by printing sliced models from our curriculum C. To maximize Equation 11 we use PPO algorithm [Schulman et al. 2017]. Each trajectory consists of a randomly selected mesh slice that is fully printed out before proceeding to the next one. One epoch terminates when we collect 10000 observations. We run the algorithm for a total of 4 million observations but convergence was achieved well before that, (Figure 21). For the training parameters we set the entropy coefficient to 0.01 and anneal it towards 0. Similarly we anneal the learning rate from 3×10^{-4} towards 0. Lastly, we picked a discount factor of 0.99 which corresponds to one action having a half time of 70 steps. This is equivalent to roughly 22 mm of distance traveled. In our training set this corresponds to 29-80 percent of the total episode length.

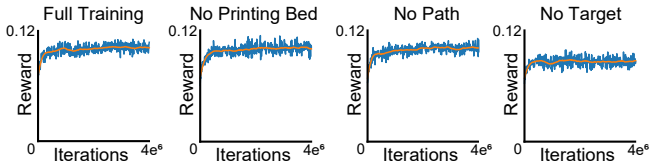


Fig. 21. Training curves for controllers with constant material flow.

We also experimented with training controllers for materials with varying viscosity, (Figure 22). In general we have observed that the change in viscosity did not significantly affect the learning convergence. However, we have observed a drop in performance when training control policies for deposition of liquid materials. The liquid material requires longer time horizons to stabilize and has a wider deposition area making precise tracing of fine features challenging.

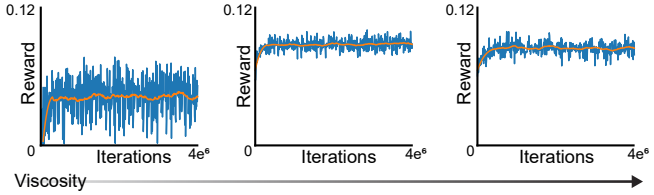


Fig. 22. Training curves for controllers with increasing viscosity in an environment with noisy flow.

Lastly, we conducted ablation studies on action space and reward function in the environment with noisy deposition, (Figure 23). We can see that employing the delayed reward had a negative effect on convergence and it is unclear if a policy of sufficient quality would be achieved.

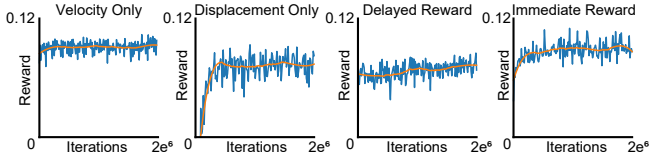


Fig. 23. Training curves for controllers with variable material flow.

For evaluation we constructed a separate dataset consisting of freeform and CAD geometries that were not present in the training. A subset of the dataset is visualized in Figure 24.

Fig. 24. Exemplar models from the evaluation dataset.

A.4 Slices Used to Estimate Outline Improvement

We estimate the quality of deposition be evaluating the under and over deposition histograms on a subset of the evaluation dataset, (Figure 25).

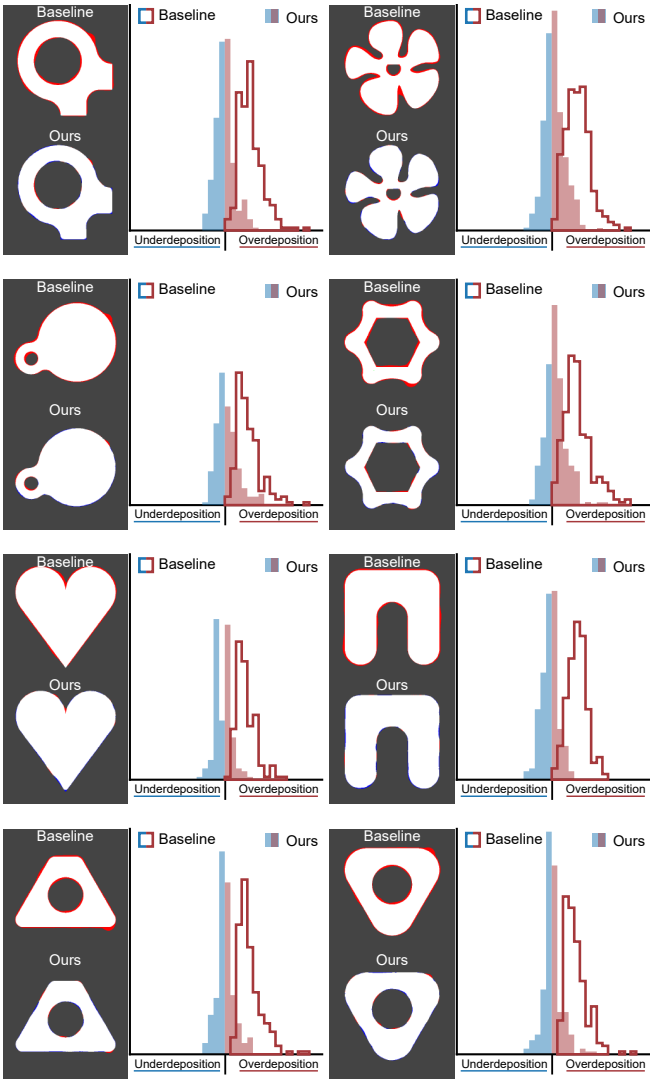


Fig. 25. Recovered histograms for a subset of slices from the evaluation dataset show tighter deposition of material achieved by our closed-loop control policy.

A.5 Detailed Physical Results

We fabricated 11 shapes from the training and 11 shapes from the evaluation dataset on a physical apparatus using a low and high viscosity material, (Figure 26).

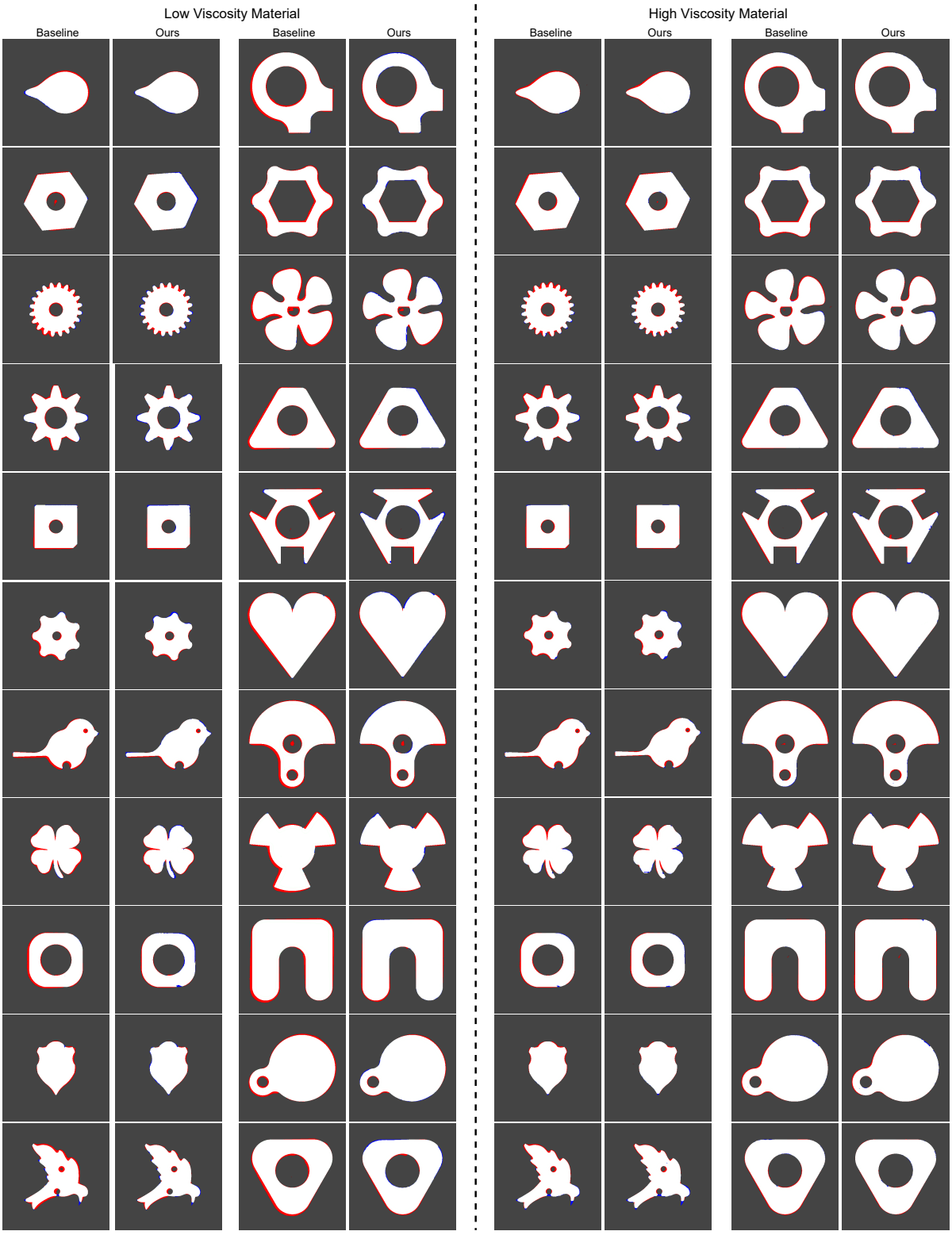


Fig. 26. Policy evaluation on physical hardware.