



# Instant flow distribution network optimization in liquid composite molding using deep reinforcement learning

Martin Szarski<sup>1</sup> · Sunita Chauhan<sup>1</sup>

Received: 5 September 2021 / Accepted: 5 July 2022 / Published online: 8 August 2022  
© The Author(s) 2022

## Abstract

Carbon fibre reinforced plastic (CFRP) manufacturing cycle time is a major driver of production rate and cost for aerospace manufacturers. In vacuum assisted resin transfer molding (VARTM) where liquid thermoset resin is infused into dry carbon reinforcement under vacuum pressure, the design of a resin distribution network to minimize fill time while ensuring the preform is completely full of resin is critical to achieving acceptable quality and cycle time. Complex resin distribution networks in aerospace composites increase the need for quick, optimized virtual design feedback. Framing the problem flow media placement in terms of reinforcement learning, we train a deep neural network agent using a 3D Finite Element based process model of resin flow in dry carbon preforms. Our agent learns to place flow media on thin laminates in order to avoid resin starvation and reduce total infusion time. Due to the knowledge the agent has gained during training on a variety of thin laminate geometries, when presented with a new thin laminate geometry it is able to propose a good flow media layout in less than a second. On a realistic aerospace part with a complex 12-dimensional flow media network, we demonstrate our method reduces fill time by 32% when compared to an expert designed placement, while maintaining the same fill quality.

**Keywords** Textile composites · Finite element analysis (FEA) · Resin transfer moulding (RTM) · Machine Learning · Optimization

## Introduction

Improved fuel efficiency of passenger flights is critical to the environmental sustainability and economic viability of air travel. In response, major aircraft manufacturers have moved to improve the efficiency of their products through new designs and new technology. The weight of an aircraft is of singular significance to fuel burn, and for this reason the industry has moved away from Aluminium as the major structural material to lighter composite materials like Carbon Fibre Reinforced Plastic (CFRP). Over the past 30 years each generation of aircraft has included more CFRP to reduce weight. Modern passenger aircraft are over 50 percent CFRP by weight, including CFRP primary structure such as the fuselage and wings on the Boeing 787 and Airbus A350 (Roberts, 2007). CFRP cure cycle time is a major driver of production rate and cost for Original Equipment Manufactur-

ers (OEMs), as well as Maintenance, Repair, and Overhaul (MRO) organisations.

In Liquid Composite Molding (LCM) techniques such as Vacuum Assisted Resin Transfer Molding (VARTM), liquid thermoset resin is infused into dry carbon reinforcement under vacuum pressure in an oven. When compared to traditional cure processes where the carbon reinforcement is preimpregnated with resin (prepreg) and then cured under pressure in an autoclave, the comparatively lower cost of ovens along with other advantages has led to increased interest in the use of VARTM in aerospace structures (Soutis, 2005). However, the time required to fill the dry reinforcement with resin means that a part produced with VARTM will spend more time in an oven than a prepreg part spends in an autoclave. Designing processes to reduce this additional cycle time will be a critical factor as production rates increase. This challenge also exists in a related family of LCM techniques which use all rigid molding surfaces allowing pressures higher than 1 atmosphere, called Resin Transfer Molding (RTM). RTM is used to produce complex 3D parts, but is limited in scalability due to the increased cost of

✉ Sunita Chauhan  
sunita.chauhan@monash.edu

<sup>1</sup> Department of Mechanical and Aerospace Engineering,  
Monash University, Melbourne, Australia

double-sided sealed tooling, and so is not applicable to larger structures.

LCM processes have two main stages (Sozer et al., 2012):

1. Placement of the dry preform on or inside the mold
2. Mold filling, in which the resin is infused or injected into the preform to saturate all the empty space between the fibers

During the mold filling step, resin is injected through inlets or 'gates', and air leaves through outlets or 'vents'. The first objective is to fully saturate the preform with resin, displacing all the air and volatile organic compounds, as not completely filling the preform leads to dry spots and defective parts that cannot be used. The second objective is to minimize the fill time. A successful LCM process will completely fill the preform quickly (Sozer et al., 2012).

There are two main approaches to reduce fill time. In a RTM setting, a high pressure can be used to increase the resin flow rate. In both RTM and VARTM settings, the design of the resin distribution system has a direct impact on how the resin flows through the preform (Hsiao and Heider, 2012) and whether the preform is entirely filled, it is therefore of primary importance to LCM process design.

A resin distribution system includes the design and location of resin injection sites, the location of air outlets, the design and location of channels or flow pipes designed to carry resin from one area of a mold to another, and the design and location of high permeability material layers ('flow media' or 'distribution media') placed on top of the preform that distribute resin preferentially (Hsiao and Heider, 2012).

The design of these systems must be done carefully and is not always intuitive (George, 2011). In industry, process engineers depend on their experience and knowledge to propose a resin distribution system design, which is then refined through trial and error loops, expensively testing fabrication strategies, rather than a systematic optimization approach. To avoid many physical experiments, Finite Element Methods (FEM) have been previously established in the literature solving Darcy's law describing the flow of resin in a dry preform (Bruschke and Advani, 1990), we briefly review the key points in Section [Simulation and Process Model](#). However, even with accurate process models, optimization of the resin distribution network still remains a challenge.

In RTM where the high pressure helps assist in achieving high resin flow rates, the design of gate and vent locations has been the subject of active research. Using various models of resin flow, many optimization techniques have been studied, we briefly review some key contributions to the RTM design optimization literature below:

- Young (1994) use a genetic algorithm to optimize 2 gate locations evaluated using a mold filling FEM that also simulates temperature. They propose an objective function weighting resin injection pressure, the maximum time difference between boundary nodes being filled, and the maximum temperature difference between nodes at the end of mold filling. They find their results are encouraging but that computational complexity is a limit to the effectiveness as the number of design variables increases.
- Mathur et al. (1999) use a genetic algorithm to optimize 2 vent locations for minimum fill time using a mold filling FEM. They propose an objective function that weights fill time and dry spot formation, and use a step function penalty to impose constraints. They report that their results make physical sense and could be used to seed a local search.
- Gokce et al. (2002) optimize the location of a single gate using a branch and bound approach and a mold filling FEM. They partition the possible gate locations into sets geometrically and exploit a locality assumption that the lower bound of the objective function (total unfilled nodes) is of the same order of magnitude as the average over a given spatial partition. They perform separate optimizations for cycle time and elimination of dry spots. The authors report that the branch and bound approach had an 86% lower computational cost when compared to a genetic algorithm, and verified it found an optimal solution via exhaustive search.
- Bocard et al. (1995) use a geometric approach based on distances which does not take into account material properties to propose gate locations for thin molds that avoid trapping air bubbles. Their approach is based on constructing geometric subdomains in the 2D plane, such that each domain contains one inlet. They then use the distance from each inlet port to the two closest points on the perimeter to determine the gate locations algorithmically, without using input from a mold filling simulator.
- Lin et al. (2000) use a quasi-newton optimization method to refine the initial placement of 3 gates using a mold filling FEM. The authors find that using a FEM with gradient based search requires either a fine mesh or adaptive meshing so that small differences in evaluated gate locations provide a useful gradient to the optimizer. They recommend that if one or more of the parameters are discrete (such as the number of gates), gradient based optimization should either not be used, or used in conjunction with a search based method like genetic algorithms.
- Wang et al. (2017) use a geometric approach (medial axis transform) to find initial symmetric designs for injection channels in RTM molds of shell-like complex parts. The initial designs are then optimized using a FEM based mold filling simulation to determine the flow pattern. This flow pattern is then geometrically partitioned to allow a

binary search to move injection channels to correct for anisotropy in the preform.

The main challenge of VARTM is the reliance on vacuum pressure. As the resin follows the pressure gradient through the preform, the varying permeability caused by complex geometries is especially prone to dry spots if the resin delivery system is not well designed. Versus RTM, there is comparatively less literature concerning optimization of VARTM resin distribution networks. We summarize some of the key approaches below:

- Hsiao et al. (2004) optimize the cross-sectional area of 2 flow runner channels, and the number of plies of distribution media in 6 sections to determine an optimal flow distribution network for a co-cured stiffened panel. The authors define an optimization objective weighting dry spot volume fraction and fill time. They use a genetic algorithm and a finite element flow model and report their GA found a solution in the top 1.3% of all possibilities, using only 15.5% of the computational time. Their results were verified by exhaustive search and experimentation.
- Kessels et al. (2007) optimize the location of 3 flow runner pipes on a complex glider shell. They use a 3D geometric mesh distance model based on the assumption that the resin first fills the nodes that are closest to the inlet, then the next closest, etc. They remark that this assumption is locally true but not globally, and suggest modifying mesh distances to capture anisotropic permeability, although this is not demonstrated. The mesh based distance model is then used in the loop of a genetic algorithm. They found generally good agreement between their model and a FEM filling model, but reported that when the flow is complex, the results differ from a FEM filling model and recommend a final calculation to refine the vent locations.
- Sánchez et al. (2015) propose a pre-design tool for resin channels on a large thin boat shell using a geometric model. The model is based on distance fields computed by level sets which are then partitioned into independent regions. The partitioning lines are the resin channel paths, resembling a medial axis. It assumes uniform and isotropic permeability in the preform and that gates should be placed on the boundary, as far away from inlets as possible. based on an assumption of uniform permeability to be used. The authors conclude that the resultant pre-design should then be computed by a full physics-based simulation.
- Sas et al. (2015) optimize 6 flow media sections on a panel and 14 sections on a complex geometry for robustness under multiple racetracking scenarios around an insert. Focusing on robustness, they explicitly enumerate the possible racetracking scenarios around the insert and use

tree-search based method to perform discrete optimization over the number of distribution media plies in each region. Of note, their tree search is able to update the number of regions if the optimization does not converge. Evaluation in the tree search is done using a FEM fill model. They report a case study in which their proposed solution is able to fill the mold for all scenarios.

Many of these approaches can be considered a multiobjective grid based placement optimization problem. A more complex and constrained variant of this general problem is heavily studied in integrated circuit design and is known as Global Placement. While there are analytical approaches like Integer Linear Programming, and numerical approaches like non-linear optimization, much like flow media placement it is often solved stochastically using GAs, Tree/Graph search, or Reinforcement Learning (Mirhoseini et al., 2020). We have focused our literature review on the optimisation techniques previously applied to flow media placement and similar manufacturing problems in composites, however for interested readers we recommend the review of the history of work in this related field that inspired ours by (Kahng, 2021).

In the case of the RTM literature, optimizations that use realistic physics-based flow models all exhibit very low dimensional design spaces (typically only one or two gate locations are optimized) and often have an intuitive starting position such as the center of the mold to work from. Conversely, VARTM resin delivery systems are often more complex, and the set of design variables is larger. Therefore most studies that consider flow pipes or channels have not used accurate physics-based models, but instead made use of symmetry and assumptions of the homogeneity of the permeability in parts, which is a significant limitation for application to VARTM aerospace structures. While the placement of distribution media has a very large effect on the flow front, natural through-thickness permeability variation in the preform can also have an effect on the evenness of the flow front along the bottom surface. Yun et al. (2017) found that this variation meant that the percentage of voids in the final filled preform increased with the permeability of the distribution media, further emphasizing that physics-based simulations with appropriate material properties are important for resin distribution network optimization.

Hsiao et al. (2004) and Sas et al. (2015) tackle the highest dimensional design spaces in a problem setting similar to ours. In Hsiao et al. (2004), the authors optimize 6 sections of resin flow media and the cross-sectional area of 2 flow runner channels, allowing multiple layers of flow media in each section, resulting in a search space with  $3^2 * 2^2 = 2916$  combinations. The main limitation of the work in Hsiao et al. (2004) is in scalability, as their proposed genetic algorithm explores a full 16% of this relatively small design space to yield the optimized result. In Sas et al. (2015) the authors

expand the search space under optimization compared to Hsiao et al. (2004) by considering up to 14 flow media sections (allowing one or zero layers of flow media at each) giving a design space of  $2^{14} = 16,384$ .

In aerospace VARTM designs with many flow media sections and multiple layers allowed at each, the design space is at least one and possibly three orders of magnitude larger than considered in Sas, and exploring that much of the design space is expensive in both computational time and real time. Designing aerospace geometry is an iterative process as requirements and interfaces are refined. Many distinct but related part designs will be proposed before the geometry is finalized. Hence, optimizations that take significant time before feedback is given to the engineers can become bottlenecks in the design process. Our proposed solution to this problem is an optimization technique that is able to transfer knowledge from previous optimizations performed on similar designs to a new design in the same way an expert would.

We achieve this by training a neural network-based agent to optimize the placement (location and thickness) of flow media through Deep Reinforcement Learning (DRL). Through experience, our agent learns the skill of placing flow media when presented with a part design. This experience encoded in the neural network can then be thought of as a skill that can be applied to other parts without requiring additional experience.

This is a shift in how to think about process optimization. Instead of directly optimizing the scenario of interest by running a process model, we train an agent on a variety of (usually randomly generated or perturbed) scenarios in order to gather relevant experience for the agent. Then, when presented with a previously unseen scenario, the agent immediately proposes new parameters without any further training. This can create significant time savings as it potentially eliminates entire optimization runs. The applicability of the agent's previous experience and therefore the overall effectiveness when the agent is presented with a new scenario will depend on the similarity between the scenarios it was trained on, and the scenario of interest.

Applying DRL to production system optimization has been recently shown to outperform conventional methods in many manufacturing domains. We recommend the review of this area by (Panzer and Bender, 2021) for a comprehensive summary. DRL has been shown to work well in the manufacturing setting through creating RL environments based on offline process model simulators such as ozonation of textiles (He et al., 2021), material draping (Zimmerling et al., 2022), metal forging (Ma et al., 2022), and brine injection (Andersen et al., 2019). The need for knowledge transfer combined with the high-dimensional decision problem and the demonstrated success of DRL in industrial process optimization motivate our use of Deep Reinforcement Learning in this domain.

Our model is trained using realistic geometries and process model runs from a design family of thin laminates with complex “pad-ups” (an aerospace term for locally thicker sections on an aerospace skin), the system is then instantly able to propose a good placement on previously unseen parts from that same family without running the process model at all. Additionally, by finetuning, the system can continually learn and improve from new experiences as it is used.

The main contributions of our work are:

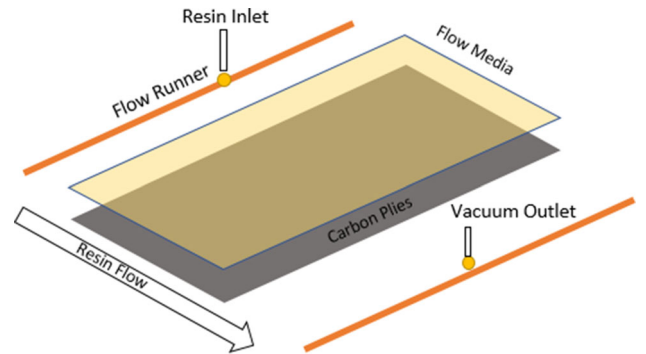
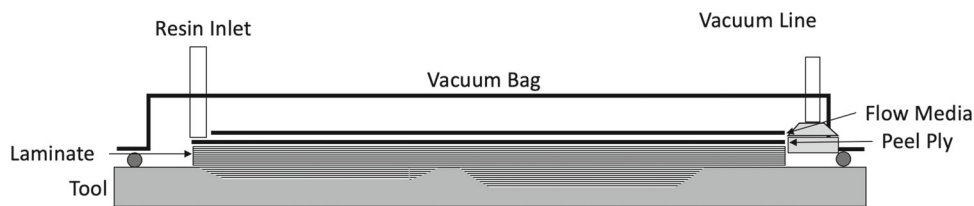
- We introduce a novel *instant* optimization method using machine learned knowledge transfer for VARTM flow media placement that is scalable to realistic aerospace parts with the objective of avoiding dry spots and minimizing fill time
- We detail our Deep Reinforcement Learning (DRL) based optimization methodology that successfully learns a skill from experience in simulation to optimize flow media placement on designs that were not seen during training, without evaluating any process model on the design under optimization
- We present a case study applying our method to a realistic aerospace laminate part with a 12-dimensional design space of size  $3^{12} = 531,441$  where we reduce fill time by 32% when compared to an expert designed placement, while maintaining a complete fill of the preform

The remainder of this paper is organized as follows. In Section [Flow distribution network optimization](#), we describe the overall resin flow optimization problem. In Section [Simulation and Process Model](#) we describe the process model used in our optimization. In Section [Flow media placement optimization](#) we describe our proposed approach to flow media placement optimization. The results and discussion are presented in Section [Results and discussion](#). Finally, Section [Limitations](#) concludes the paper and discusses future work.

## Flow distribution network optimization

In this study we consider a VARTM process occurring in an oven with metallic tooling underneath the CFRP and soft tooling on top as in Fig. 1. The tooling is Inner Mold Line (IML) controlled as is used in complex structures in the aerospace industry (Hiken, 2017), meaning complex pad-ups are placed on the tool side to control assembly tolerances. A stack of carbon sheets (plies) are stacked (layered up) between the top and bottom layer of tooling materials. In VARTM, the soft tooling materials are sealed to the mold surface, and the carbon plies are then held under vacuum to compact them and remove air. The resin is injected through inlets and leaves through outlets under 1 atmosphere of vacuum, this pressure

**Fig. 1** VARTM tooling arrangement



**Fig. 2** Example resin distribution network containing a flow runner and flow media

differential pulling the resin through the dry preform and filling the empty space between fibers with the resin. Once saturated, the now filled preform is held under vacuum and the temperature is raised to cure the resin and create the final composite part. In order to ensure the preform is appropriately filled, a resin distribution network is used to direct the flow of the resin during infusion.

The resin distribution network consists of the inlets, outlets, flow channels or pipes, and high permeability distribution media (flow media) layers. As the impregnation of the resin into the dry reinforcement is achieved only through vacuum, the resin distribution network design significantly influences part quality and cycle time.

While inlets can be single points, it is typical to use a 'flow runner' (channel or leaky pipe) so that injected resin flows from a line instead of a point. These are commonly placed along one edge of the part for practicality. Default practice is to cover most of the top surface of the preform with a layer of high permeability distribution media in order to encourage resin flow across the top of the preform (in-plane) first, and then allow the resin to flow through thickness. Due to the vacuum outlet location on the opposite edge from the inlet, the pressure differential pulls resin both in-plane and through thickness, however as the permeability through-thickness is much lower than the in-plane permeability, care with the placement of distribution media must be taken to ensure the resin doesn't flow across the top of the preform in-plane before filling through-thickness. An example of a resin distribution network containing a flow runner and flow media is displayed in Fig. 2.

While filling, the resin flow front will be non-uniform through the thickness of the preform. This phenomena is displayed in Fig. 3. In aerospace parts with large scale and/or complex geometries with locally varying thicknesses, this lag in the flow front between the top and bottom surfaces is exacerbated, and can cause the resin to 'racetrack', arriving at the outlet through a more permeable path around thick areas instead of through them before the whole preform is full, trapping off vacuum and causing dry spots due to resin starvation (known as 'trapoff'). An illustration of this phenomena for a part with varying thickness as an example is displayed in Fig. 4).

As a preform that is not almost completely filled will lead to a defective part, a sub-optimal resin distribution network design causes both quality issues and poor cycle time.

Optimizing resin distribution is therefore an important problem in industry, however current practice relies on manual optimization by trial and error either through simulation physical experiments.

### Simulation and Process Model

In the literature, flow optimization makes use of either a numerical process simulator (Gokce et al., 2002; Lin et al., 2000; Young, 1994; Boccard et al., 1995; Mathur et al., 1999; Hsiao et al., 2004) or a geometric surrogate model (Sánchez et al., 2015; Kessels et al., 2007), in order to evaluate an objective function without doing expensive physical experiments.

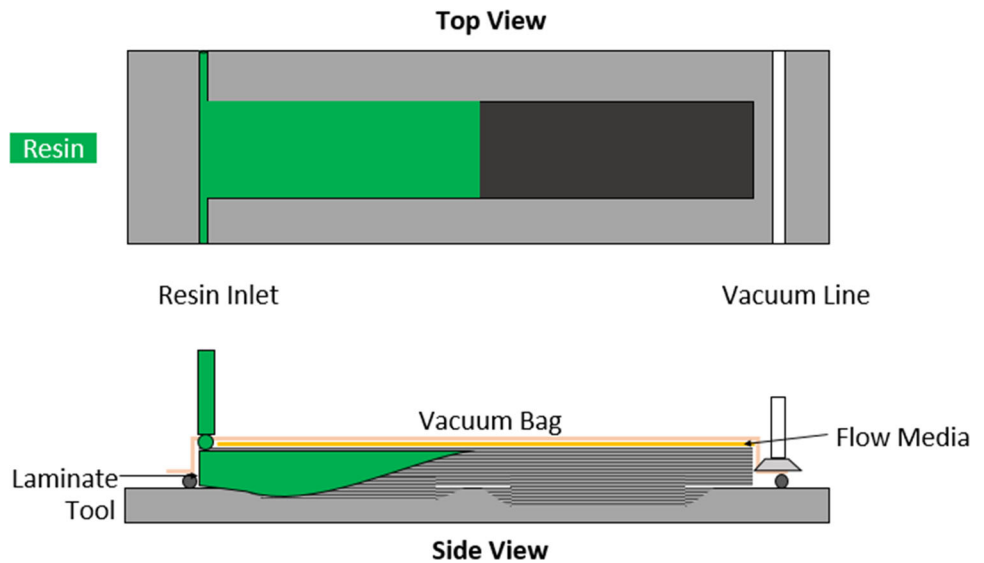
Mathematical models of the relevant physics of resin flow during processing of thermoset composites and their solutions using Finite Element Methods have been previously established in the literature (Bruschke and Advani, 1990) and the key points briefly reviewed here. We do not propose a new process model in this work, but use existing work to feed our optimization.

When filling the dry preform, we can treat the resin as a fluid flowing through an anisotropic porous medium and we can describe it with Darcy's law (Eq. 1).

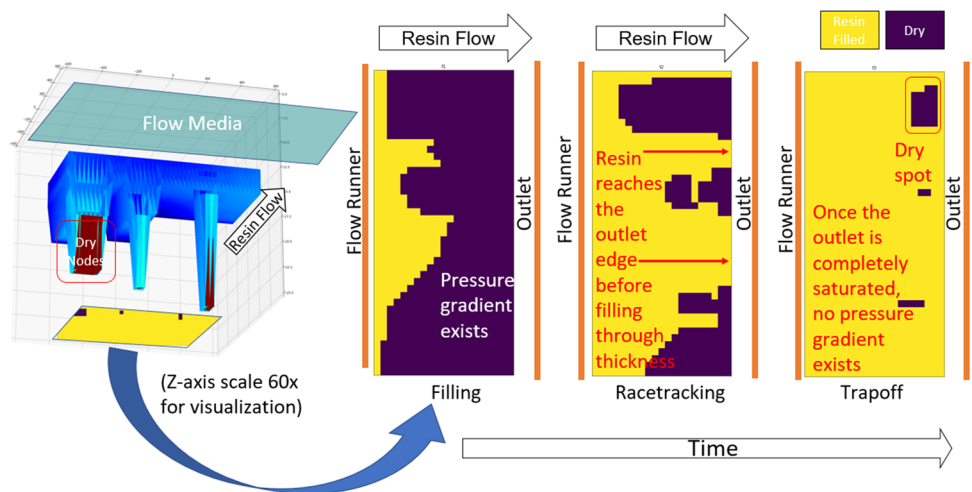
$$\vec{u} = -\frac{K}{\eta} \nabla P \tag{1}$$

Where  $\eta$  is the viscosity of the resin,  $K$  is the permeability tensor of the preform,  $\vec{u}$  is the average velocity, and  $\nabla P$  is the pressure gradient in the flow field.

**Fig. 3** Non-uniform through-thickness resin fill front showing resin flowing quickly along the distribution media faster than through the thickness of the preform



**Fig. 4** Visualization of ‘trapoff’



Darcy’s law states that the velocity of a fluid flowing through a porous medium is directly proportional to the driving pressure drop.

If we assume that the resin is incompressible and the flow is quasi-steady state in the domain (the domain is at full saturation) as in Eq. 2, we generate the following PDE (Eq. 3).

$$\nabla \cdot \vec{u} = 0 \tag{2}$$

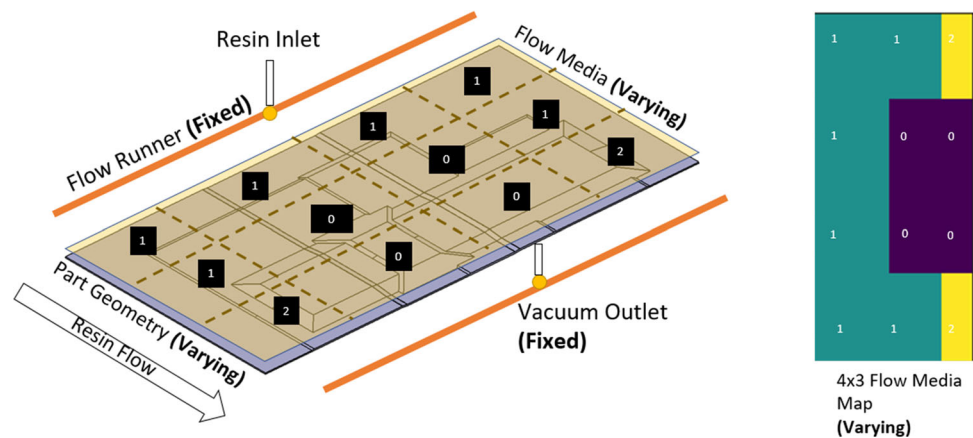
$$\nabla \cdot \left( \frac{K}{\eta} \nabla P \right) = 0 \tag{3}$$

This second order PDE can be solved when the boundary conditions (no flow through the boundary, inlet at the process injection pressure, and flow front pressure at zero) are prescribed. This is a moving boundary problem as the resin domain changes over time. However, once the velocity is obtained at the flow front, we can project the expansion of the resin saturated domain for the next time step, and update the

pressure distribution. This is discretized using finite element-control volume (FE-CV) methods to perform these updates.

As suggested in (Hsiao et al., 2004), by assigning different (but fixed) permeability values (in 3D) to the preform, flow runner, and flow media elements, we can model the resin flow during the VARTM process for the purposes of flow media layout optimization. We use the validated FE-CV model formulation from Brusckie and Advani (Brusckie and Advani, 1990) which has been used with good correlation to experiments in (Mathur et al., 1999; Gokce et al., 2002; Maier et al., 1996; Hsiao et al., 2004). While we use this previously validated approach in our study, we emphasize here that our optimization method detailed in the next section makes no particular assumptions on the process model except for access to the element fill volumes and times. With the appropriate software engineering it could be applied to work with other tools such as PAM-RTM™ (ESI-Group™), and LIMS (Maier et al., 1996).

**Fig. 5** Flow Media parameterization into a discrete 4 x 3 grid for optimization



## Flow media placement optimization

We consider the problem of determining optimal placement of flow media for the fabrication of thermoset-matrix composites using a VARTM process. Our optimization target is the family of thin laminate parts with complex pad-ups as are typically found on aircraft control surfaces. The objective is to minimize the cycle time while simultaneously satisfying the primary manufacturing constraint that all of the part is filled with resin.

We assume the part is held at a constant infusion temperature at all locations during filling. The resin delivery network under optimization consists of flow media layers placed on the top surface of the preform. In all our scenarios there is a flow runner channel running along one side of the tool, with a vacuum outlet on the opposite side with the runner, inlet, and outlet locations all held fixed. For the purposes of optimization, we discretize the flow media design space into a 4 by 3 grid, each position of which can accept 0, 1, or 2 layers of flow media. A visualization of the parameterization is presented in Fig. 5. The Flow Media Map is a 2D matrix containing the integers 0, 1, or 2 at each coordinate  $(i, j)$  which corresponds to the number of flow media layers at that position in the grid. This 1:1 correspondence between the Flow Media Map matrix and the flow media arrangement is clear by inspecting Fig. 5. This parameterization results in a design space of  $3^{12} = 531,441$  combinations. The grid discretization of 4 by 3 was chosen to be at least an order of magnitude larger than considered by previous literature while not exceeding reasonable manufacturing complexity, however our optimization method does not depend on the exact discretization.

We use two-dimensional elements to represent the flow distribution media in the finite element model with constant thickness, porosity, and permeability. To model the addition or removal of flow media, the Finite Element Mesh used as input is modified by changing the thickness and therefore control volume of the flow media elements that are mod-

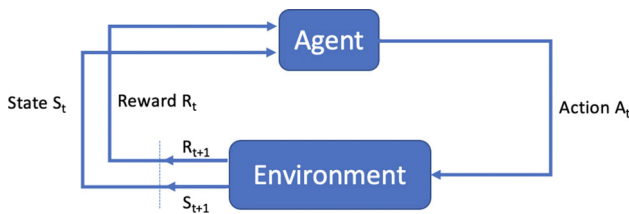
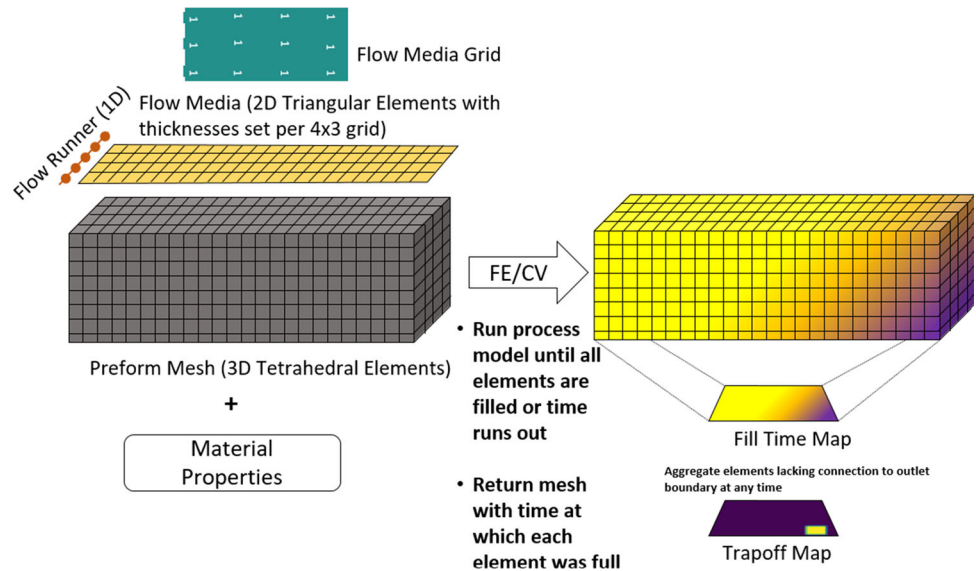
eled on the top of the preform. The part is meshed in 3D with tetrahedral elements and the flow progression is solved as described in Section [Simulation and Process Model](#). The process model stores the time at which each element is full which allows computation of flow fronts at any time  $t$ . This output is used to produce the Fill Map (a heatmap where the values represent the time at which the bottom surface elements fill) and the Trapoff Map (a binary map where all values are zero except element locations lacking connection to the outlet boundary at any time). These maps can be interpreted as images produced by virtual cameras if the tooling surface was transparent, as is common in validation studies. This information gives the agent spatial context with which to make decisions on flow media placement. Section [State](#) describes how the process model information is encoded into the state consumed by the agent. A visualization of the process model elements used in optimization is presented in Fig. 6.

## Reinforcement learning

Deep Reinforcement Learning (DRL) approaches have recently enjoyed success in solving control problems with high-dimensional inputs from simulators where control policies are difficult to model formally. Relevant examples are playing Atari from pixels (Mnih et al., 2015), learning object manipulation or locomotion in robotics, and playing strategy games like Go (Silver et al., 2016) and StarCraft (Vinyals et al., 2019). In industry, these approaches have successfully been used to control HVAC systems in datacenters (Moriyama et al., 2018), allocate computing resources (Mirhoseini et al., 2017), and design semiconductors (Mirhoseini et al., 2020). In composites, DRL was used to optimize VARTM temperature profiles and tooling thicknesses in (Szarski and Chauhan, 2021).

We note that the placement of resin distribution network have similarities to circuit placement problems in semiconductor design. We are inspired by the industrial success in the

**Fig. 6** The 3D, 2D, and 1D mesh elements model the dry preform and resin distribution network. Once the process model is run with the resin material properties, the output maps are generated for use by the optimizer



**Fig. 7** Reinforcement learning setting

area of data-driven design of these complex systems using Reinforcement Learning. Of particular interest is the ability of neural network ‘agents’ trained via reinforcement learning to generalize from accrued experience and apply the knowledge gained to new scenarios immediately without further training.

In the setting of Reinforcement Learning (RL), we have an agent interacting with an environment in discrete time steps. At each time step  $t$ , the agent receives the environment’s current state  $S_t$ , and the agent must choose an appropriate action  $A_t$  in response. After the agent executes the action, the agent receives a reward  $R_t$  and a new state  $S_{t+1}$  (See Fig. 7). We refer to the sequence  $(S_t, A_t, R_t, S_{t+1}, A_{t+1}, R_{t+1}, \dots)$  (i.e. the history of what action the agent took in each state and the subsequent reward it received), as the *trajectory*  $\tau$ . Informally we can think of this as the experience the agent is able to learn from.

The goal of Reinforcement Learning is therefore to train an agent so that it “knows what to do”. This is equivalent to learning the best action to take in a given state, so as to maximize the expected numerical reward over time. We refer to the map between states and actions as the *policy*.

In this setting, each time step  $t$  represents one full simulation with a set of input process parameters proposed by the agent. The reward function is calculated on the output,

and over time the agent learns the relationships between the input parameters and the output reward in order to make better choices of process parameters based on its experience.

Markov Decision Processes are a mathematical framework for modelling reinforcement learning problems. For a thorough background we recommend the treatments in (Sigaud and Buffet, 2013) and (Sutton and Barto, 2018). However, we review the definition here.

Formally, we consider the flow media placement problem as an infinite-horizon discounted Markov Decision Process (MDP). An MDP  $M$  is a tuple  $M = \langle S, A, P, R, \gamma \rangle$ , where:

- $S$  is a set of states (e.g. a spatial representation of the flow media placement, part geometry, and flow simulation output),
- $A$  is a set of possible actions to control the system (e.g. adding or removing flow media from locations on the tool),
- $P : S \times A \times S \rightarrow [0, 1]$  is the state-action-state transition probability distribution (e.g. what is the impact of a change to flow media placement on the flow simulation)
- $R : S \times A \times S \rightarrow \mathbb{R}$  is the reward function for transitions (e.g. fill time and trapoff penalties)
- $\gamma \in [0, 1)$  is a discount factor for future rewards

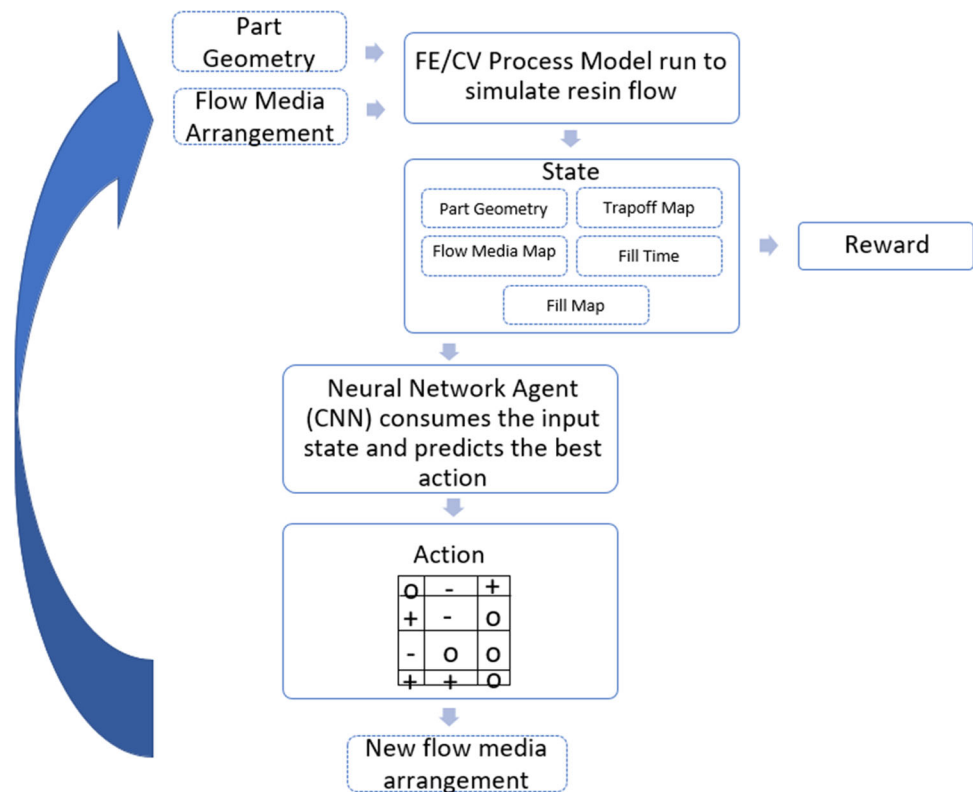
We seek a policy  $\pi : S \times A \rightarrow [0, 1]$  that maximizes the discounted expected return  $\eta$  (i.e. the reward expectation over multiple timesteps).

$$\eta(\pi) = \mathbb{E}_\tau \left[ \sum_{i=0}^{\infty} \gamma^i R(s_i, a_i, s_{i+1}) \right] \tag{4}$$

With  $\tau = (s_0, a_0, s_1, a_1, \dots)$ . With flow media placement posed as an MDP, the optimization problem is equivalent



Fig. 8 A single timestep of RL



to finding the policy  $\pi$  that maximizes a reward function designed to minimize trapoff and total fill time. A critical point is that our policy  $\pi$  must be learned from experience with state-action pairs (i.e. the expected fill pattern of a given placement on a given geometry) that have varying geometry, such that the agent can learn a skill that generalizes to similar parts. In our flow placement formulation we hold the flow media grid size, material properties, and inlet/outlet arrangement fixed. During training, we refer to a single trajectory  $\tau$  as an *episode*. This means that every timestep  $t$  from the RL agent's point of view is a complete run of the FEM and therefore distinct from the time variables in the PDE. As the action at each timestep adds or subtracts a single layer to each grid cell, a single timestep is not enough for the flow media grid to reach our maximum thickness of two layers per cell. Due to this and the stochastic nature of many RL algorithms, we model the decision problem as an MDP with episodes of 2 timesteps, which allows grid cells to reach the maximum thickness in our problem setting. To enable more extreme designs in future work, the episode length could simply be increased with no change to the model. Every episode, a new part geometry is generated, such that the agent receives experience during training across a variety of flat laminates with local pad-ups (see Section [Random part mesh generator](#)).

### Flow media placement reinforcement learning environment

We implement the environment following the Open AI Gym interface (Brockman et al., 2016), which requires `reset()` and `step()` methods, which reset the simulation or progress it one timestep, respectively. Because it is necessary to fully run a flow model with a flow media placement in order to calculate a useful reward, in our case each timestep of the environment reflects a full run of the flow model with a given placement, with the agent modifying the placement each time. A flowchart of this process is shown in Fig. 8.

We defined and implemented a resin flow environment for reinforcement learning as follows:

#### State

- 64x64x1 Part Geometry Height Map
- 64x64x1 Part Fill Map
- 64x64x1 Flow Media Map
- 64x64x1 Trapoff Map
- 64x64x1 Absolute Fill Time Channel

As our policy neural network is designed to work with 2D data, our proposed state representation is image based. After each run of the process model we produce 5 images of resolution 64 x 64 for each process model run as input to the policy network, representing the part geometry, flow media layout, and the resulting resin flow. All maps are normalized to the range of [0.0, 1.0] by the environment when they are given to the DRL agent. The Part Geometry Height Map is a 2D view of the part thickness normalized to 25.4mm thick. The Part Fill Map is a 2D heatmap representing the latest fill time of a preform element on the tooling surface at each point, normalized against the maximum fill time of any element in the map. The Flow Media Map represents the spatial flow media density as solid blocks as in Fig. 5 and is normalized against the maximum allowable flow media thickness. The Trapoff Map is calculated as follows:

1. Step through the flow progression in  $i$  steps of  $1/100$  the maximum fill time, calculating the area  $A_i$  of all tooling surface preform elements that are not connected to the boundary at step  $i$
2. Take  $i$  such that  $A_i$  is at its maximum
3. Create an image such that all tooling surface preform elements that are not connected to the boundary are set to 1 and the background is set to 0

The trapoff map can be seen as a spatial representation of the areas that have high trapoff risk (unfilled nodes with no connection to the boundary at any time). Both the trapoff map and fill map are produced with a virtual surface camera looking at the tooling surface elements as in Fig. 9.

The entire state representation is displayed in Fig. 10.

**Action**

The flow media placement optimization problem is treated as discrete action control, with the agent choosing to add, remove, or leave flow media layers at all locations on the tool at every timestep through a multinomial action distribution of dimension  $n * 3$ , where  $n$  is the number of discrete locations where flow media may be placed on the preform.

- $n$  commanded placement modifications for flow media thickness [Remove, Leave, Add]

The flow media actions are applied as in Fig. 11. We use the flow media map color convention for the remainder of the paper.

**Reward**

The reward at each step is a fill time penalty based on the maximum fill time of any element in the laminate:

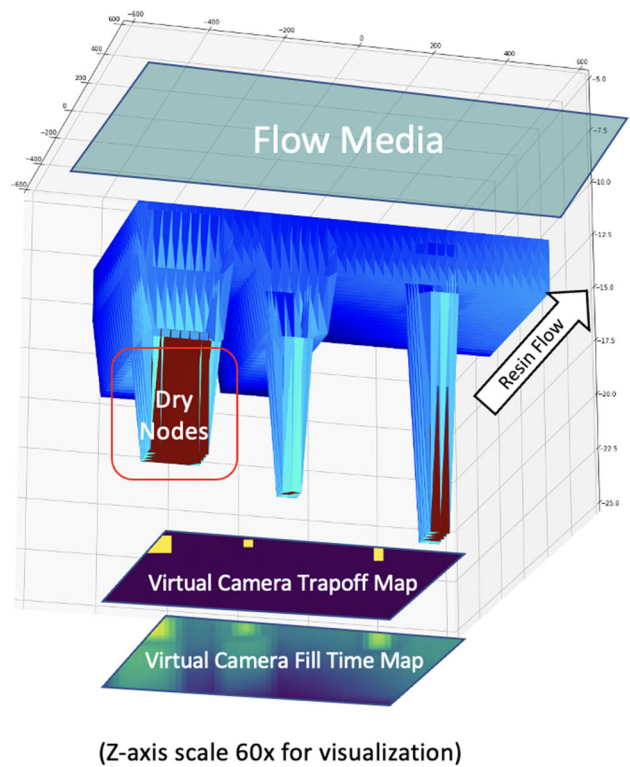


Fig. 9 Production of trapoff and fill maps

$$r_T = - \frac{\max_i t_i}{C} \tag{5}$$

where  $C$  is a fill time normalization constant set to the largest allowable fill time,  $t_i$  is the fill time for element  $i$ , and  $i$  enumerates all mesh elements that belong to the preform.

The time penalty is added to a penalty considering all preform elements not connected to the border (trapoff):

$$r_F = - \sum_{f=0}^{f=F} \frac{w * |E \setminus (E \cap B)|_f}{n} \tag{6}$$

where  $f$  is the step in the FEM flow simulation,  $F$  is the final step of the FEM flow simulation,  $|E \setminus (E \cap B)|_f$  is the number of elements that are not path connected to the boundary at time  $f$ ,  $n$  is the number of mesh elements that belong to the preform, and  $w$  is a weighting factor for disconnected element count that is set to 4 in our experiments. The trapoff map considers the entire fill time and not just the final state to avoid sensitivity to extremely long fill times - as the pressure in the preform is unlikely to ever be exactly zero it is possible that in the model, given a large amount of time, these elements may eventually fill, but this may not be physically realistic and is almost certainly not useful.

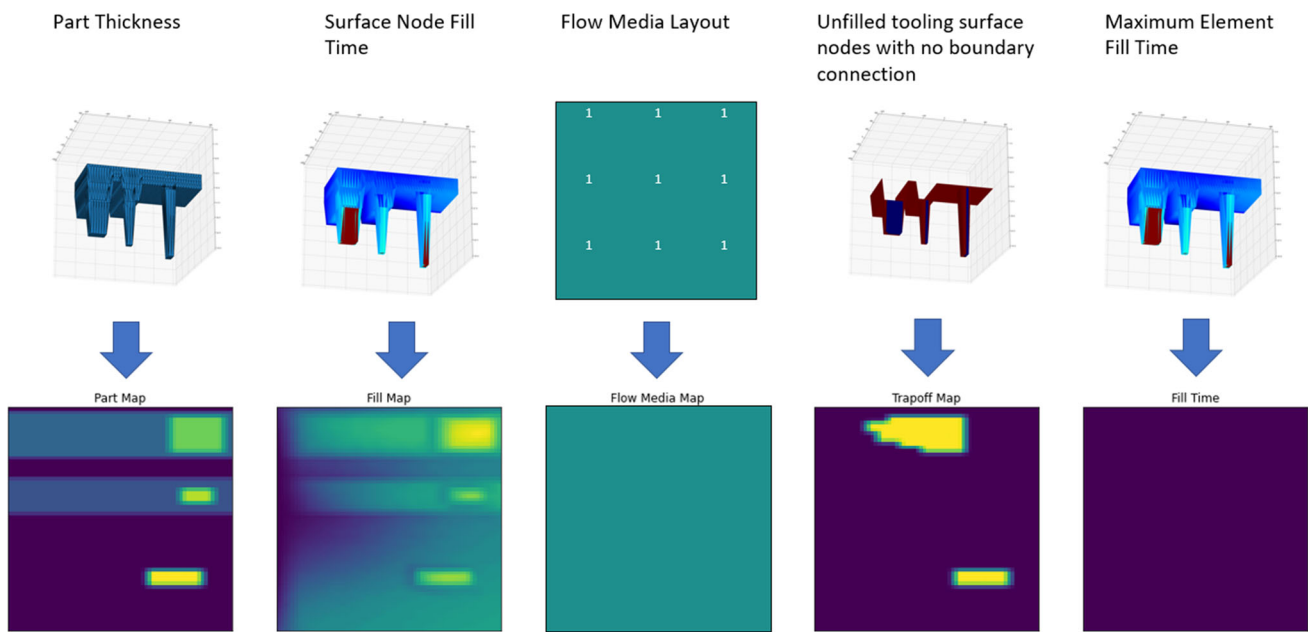
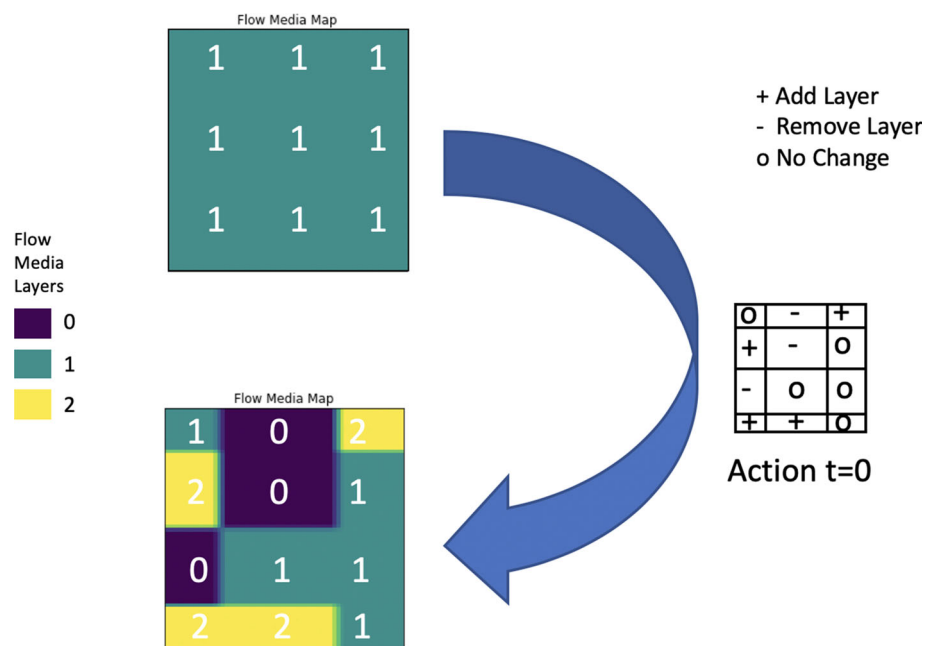


Fig. 10 State Representation

Fig. 11 Action application



The total reward at each timestep is designed to balance fill time and trapoff rewards through a weighted average:

$$r = 0.5 * r_T + 0.5 * r_F \tag{7}$$

**RL Interaction Example**

In the beginning of training, actions taken by the RL agent in each episode will appear random as the agent explores, as in Fig. 12. Each time the process model is run, the reward is calculated, penalizing trapoff and fill time. During training,

the agent learns a policy that can take in the 5 maps as the current state, and propose an action that is likely to lead to an increased reward (faster fill time and/or less trapoff). A schematic of an episode later during training is displayed in Fig. 13. In this episode, presented with different random geometry, we can see the agent discovers an 'L-shaped' flow media placement strategy for this pad-up geometry and then refines it. This experience can be used by the agent later, on further geometries.

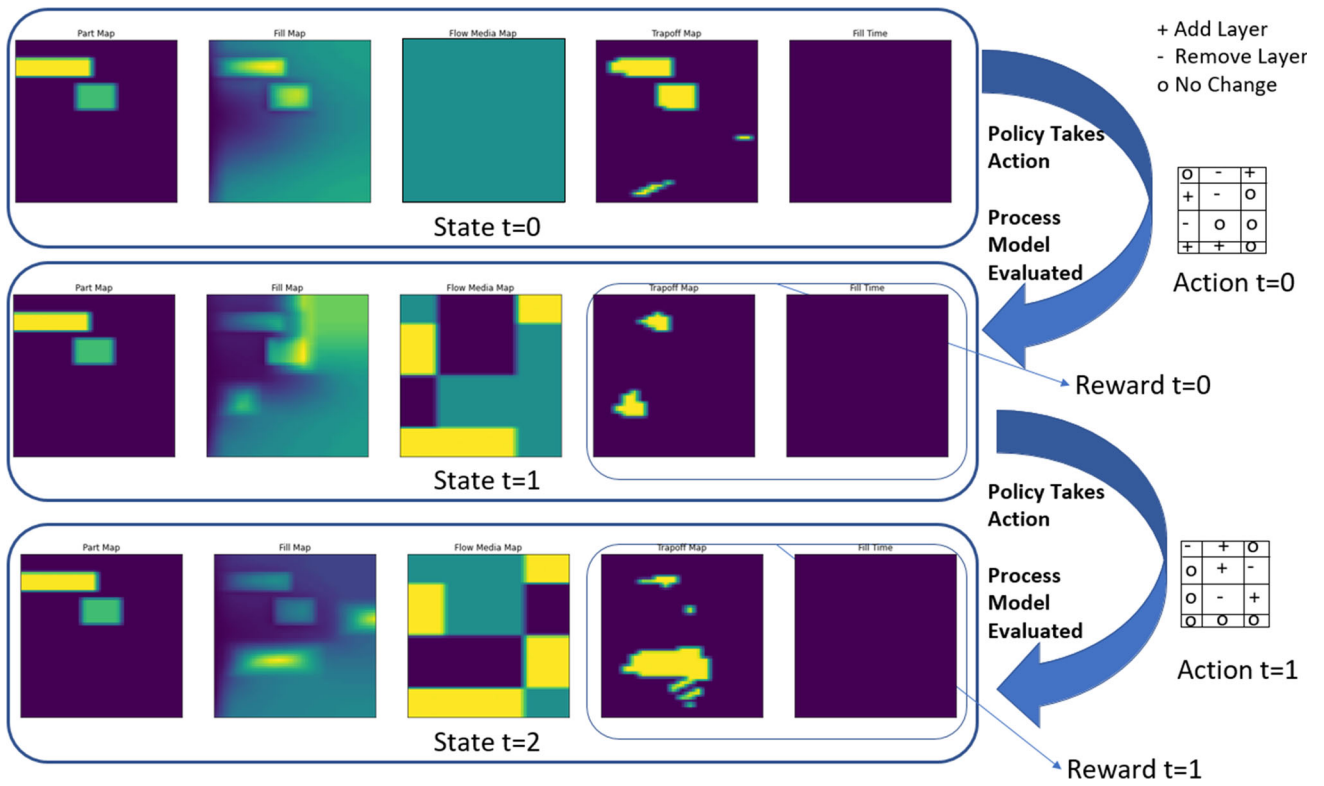


Fig. 12 One episode at the start of reinforcement learning

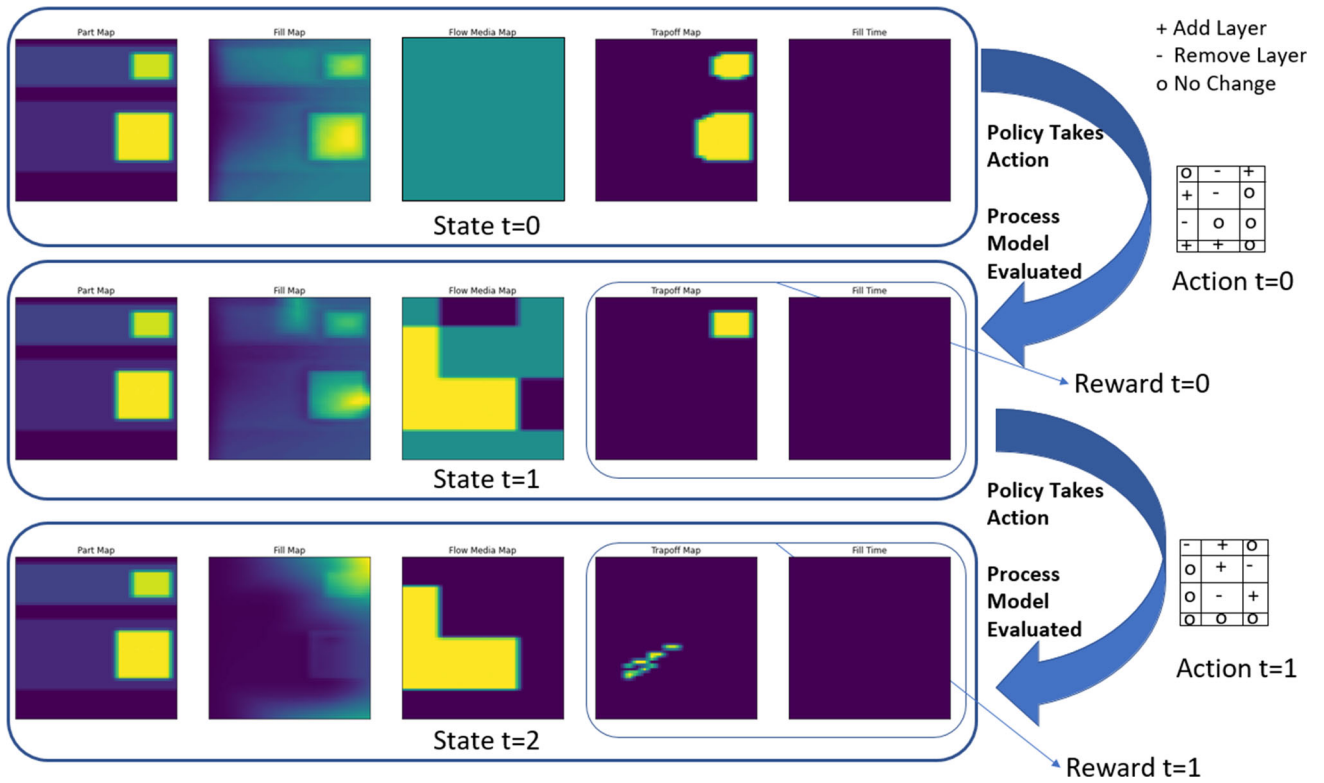


Fig. 13 One episode later during reinforcement learning

### Proximal policy optimization

Our policy  $\pi_\theta$  mapping states to actions is a Convolutional Neural Network that consumes the 5 2D Matrix heatmaps stacked as a 5-dimensional state tensor (see Section State) and outputs the mean  $\mu$  and standard deviation  $\sigma$  vectors of a 12 (3x4) dimensional multivariate gaussian distribution which determines the next placement actions (remove, leave, or add, for each of the 12 grid locations). Informally, training the policy involves collecting samples of the reward given when the agent takes a certain action in a given state, and then based on the reward, modifying the agent’s policy to make it more or less likely that the agent takes that action when it is presented with that same state in the future. The power of the Convolutional Neural Network policy is that it can generalize between similar states, so that not every state needs to be previously observed in order to take a good action.

In the context of flow media placement, during training the agent observes the results of its many different flow media placements on many different part geometries, and uses this experience of how part geometry, flow media, and fill patterns interact to increasingly make better placements (as measured by the reward function). Each simulation of (geometry, placement design, fill pattern) is a state-action-reward trajectory that is used to train the policy.

In order to train the policy neural network, we use the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm. PPO is a policy gradient approach using advantage estimation to smooth out the gradient.

Formally, we denote a policy  $\pi$ , a state action function as  $Q_\pi$ , the value function as  $V_\pi$ , and the advantage function as  $A_\pi$ . PPO updates the parameters  $\theta$  of the neural network policy  $\pi_\theta$  by taking multiple steps of minibatch Stochastic Gradient Descent (SGD) over collected state-action-reward trajectories and solving Eq. 8.

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \mathbb{E}_{\mathbf{s}, \mathbf{a} \sim \pi_{\theta_k}} \left[ \min \left( \frac{\pi_\theta(\mathbf{a}|\mathbf{s})}{\pi_{\theta_k}(\mathbf{a}|\mathbf{s})} \right) A_{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a}), g(\epsilon, A_{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a})) \right], \tag{8}$$

Where:

$$g(\epsilon, A) = \begin{cases} (1 + \epsilon)A & A \geq 0 \\ (1 - \epsilon)A & A < 0 \end{cases} \tag{9}$$

$$A_\pi(\mathbf{s}, \mathbf{a}) = Q_\pi(\mathbf{s}, \mathbf{a}) - V_\pi(\mathbf{s}) \tag{10}$$

$$Q_\pi(\mathbf{s}_i, \mathbf{a}_i) = \mathbb{E}_{\mathbf{s}_{i+1}, \mathbf{a}_{i+1}, \dots} \left[ \sum_{l=0}^{\infty} \gamma^l R(\mathbf{s}_{i+l}, \mathbf{a}_{i+l}, \mathbf{s}_{i+l+1}) \right] \tag{11}$$

$$V_\pi(\mathbf{s}_i) = \mathbb{E}_{\mathbf{a}_i, \mathbf{s}_{i+1}, \dots} \left[ \sum_{l=0}^{\infty} \gamma^l R(\mathbf{s}_{i+l}, \mathbf{a}_{i+l}, \mathbf{s}_{i+l+1}) \right] \tag{12}$$

## Results and discussion

### Experimental setup

#### RL policy network

The policy network consists of a Convolutional Neural Network (CNN) feature extractor based on (Mnih et al., 2015), with 3 Convolutional layers (8x8, 4x4, and 3x3 kernel sizes, with strides of 4, 2, and 1 respectively) and 2 Linear layers with ReLU nonlinearities between all layers. The architecture of the feature extractor is displayed in Fig. 14. At each environment step, the policy network outputs a mean and variance vector of a multivariate gaussian representing the action to take at each location on the tool (add, remove, or leave flow media layers). Our training was purely model-free, based on abstract representations of states and actions, no knowledge of the placement problem was encoded into the policy network.

#### Simulation environment

The environment was made up of a random part mesh generator and a 3D finite element model solving for the flow progression. The material properties used in the simulation were based on a T700 fiber based woven fabric and RTM6 resin, and are displayed in Table 1. Each step of the environment runs the FEM with the new distribution of flow media until the part is either filled or a iteration limit is reached. Total training time is about 12 hours on a 72 cores of an Intel Xeon 2.0Ghz CPU.

#### Random part mesh generator

To learn a generalizable agent we seek to train on realistic examples that could have been drawn from the distribution of real parts of interest. Given the iterative process involved in aerospace design, we restrict our random generator to a single part type, with the aim of training an agent that is able to generalize to modified designs within that part type. The geometry family of interest has complex pad-up regions in the tooling surface, with a flat surface at the top on which flow media is placed. The laminates are all 1125mm x 525mm. We create a simple generative model of a skin-like laminate with thick pad-ups similar to those used at interfaces on control surfaces.

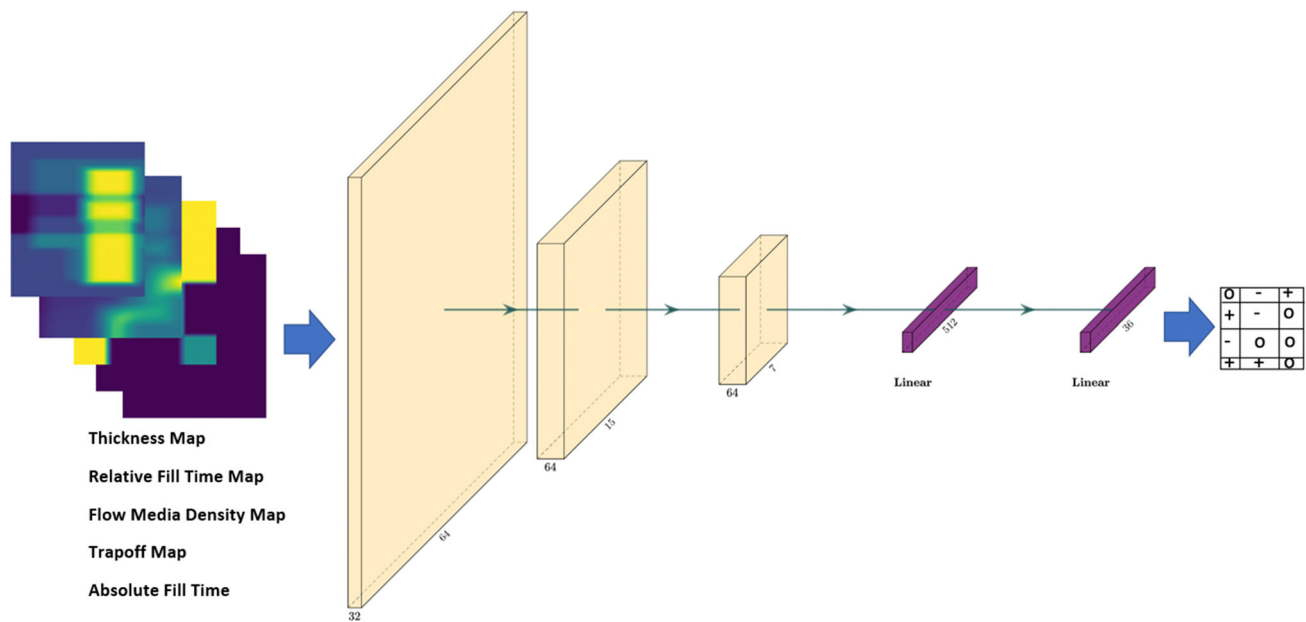


Fig. 14 Neural network architecture

Table 1 Process model parameters

Parameter	Value	Unit
Flow Runner Permeability	$1 \times 10^{-5}$	$m^2$
Flow Media Permeability	$1 \times 10^{-6}$	$m^2$
Flow Media Fiber Volume Fraction	50%	-
Flow Media Thickness	0.001	m
Preform Permeability $x_x$	$1 \times 10^{-9}$	$m^2$
Preform Permeability $y_y$	$1 \times 10^{-9}$	$m^2$
Preform Permeability $z_z$	$1 \times 10^{-11}$	$m^2$
Preform Fiber Volume Fraction	17%	-
Pressure Gradient	1	atm
Resin Viscosity	59	mPa.s

The generator works as follows:

- Generate 2-3 random, non-overlapping intervals along the panel y-axis
- Apply a random thickness to these intervals within [3.175, 6.35] mm
- Within each interval, generate a thick pad-up between [10,15] mm

We sample 10 random panels at the start of training, and use the same 10 thereafter. Examples of two random panels are displayed in Fig. 15. Thickness maps of 10 panels sampled during training are available in Appendix A.

## RL algorithm implementation

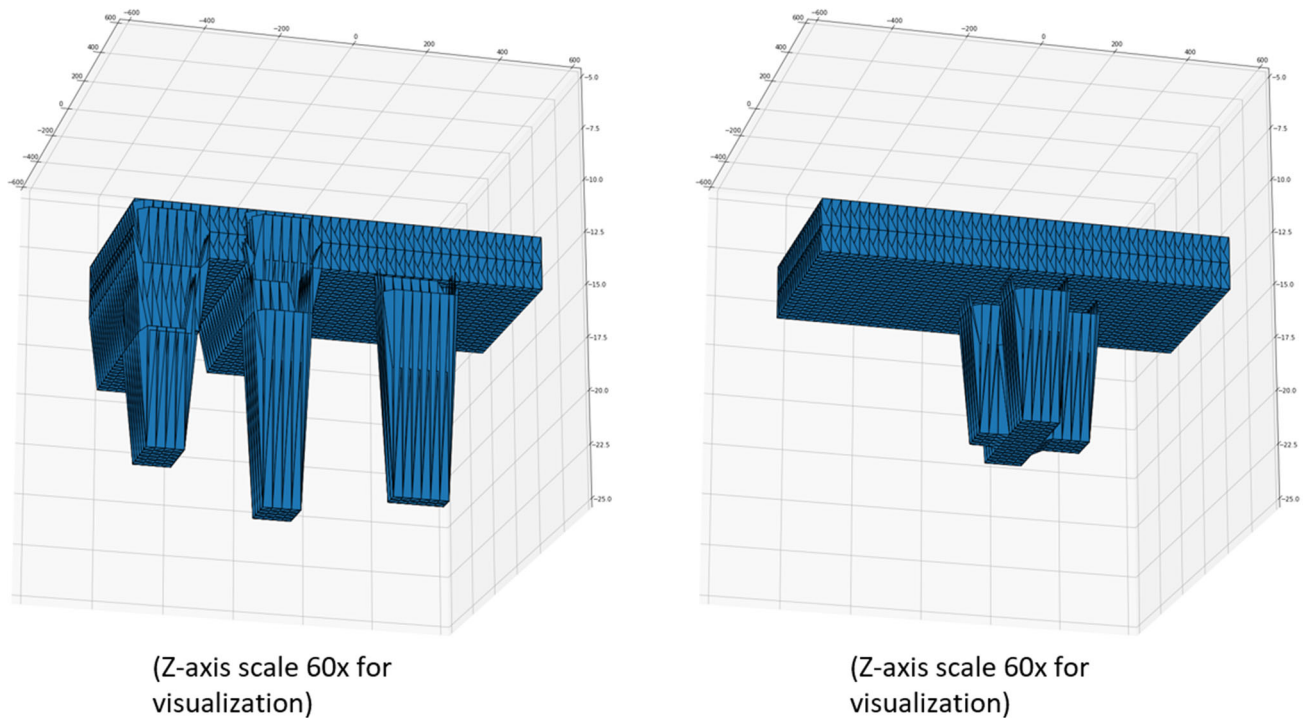
We use the open-source Stable Baselines 3 (Raffin et al., 2019) implementation of the PPO algorithm in PyTorch, distributed across 72 processes using the Message Passing Interface (MPI) (Dalcin et al., 2011) for Python to allow efficient communication between processes. The environment is designed according to the Open AI Gym interface (Brockman et al., 2016), and wrapped in the OpenAI Baselines *Vec-Normalize* wrapper, which computes a running average of observations and rewards for use during learning.

## Experiments

We based our training hyperparameters on known-good values from (Schulman et al., 2017). As the environment is deterministic and two successive actions can maximize or minimize flow media in all locations, the Steps per Episode was reduced to 2 in order increase the variety experienced by the agent. The following hyperparameters were used in our experiments:

- Learning Rate = 0.0003
- GAE  $\lambda = 0.95$
- Batch Size = 48
- Minibatch Size = 48
- Steps per Episode = 2
- Episodes = 16000

We train our agent only on 10 randomly generated panels as described in Section [Random part mesh generator](#). As in



**Fig. 15** Randomly generated laminate examples

Section [Reinforcement learning](#), during the training process a panel is randomly selected every episode, so that the agent receives experience across a variety of panels. We evaluate the ability of our agent to use the knowledge it has gained during training by asking it to instantly optimize two panels it has never seen before but are from the same family (thin laminates with pad-ups of dimension 1125 mm x 525 mm.). We use two geometries for this purpose, a randomly generated geometry with two large pad-ups, and a realistic human designed geometry from the control surface family of interest. We reiterate here that the case studies are performed without any training, and that the results for these panels are instant.

The two geometries in our case study are displayed in [Fig. 16](#).

## Results

### Randomly generated test part

[Figures 17 and 18](#) present the pre- and post- optimization states of the Randomly Generated Test Part, respectively. In each case the 'pre-optimization' state represents a uniform distribution of one layer of flow media placed across the laminate, which is the default used industrially for flat laminates and unlikely to be optimal when applied to complex geometries. Before optimization, only 88% of the part was filled due to

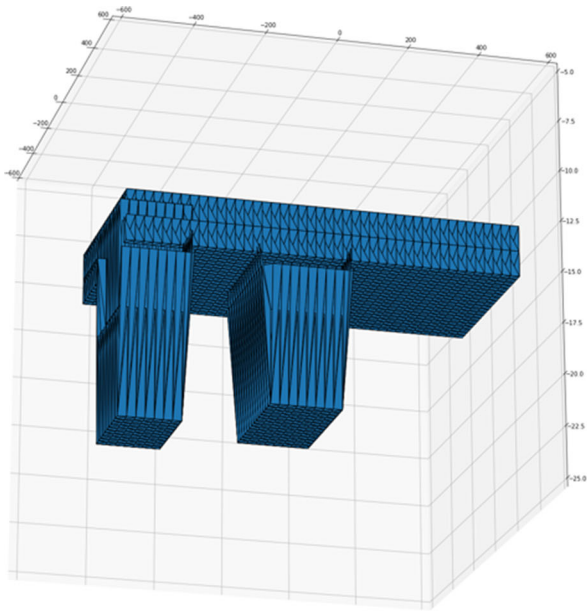
trapoff around the thick pad-up areas, after flow media layout optimization, 99.9% of the part is filled.

### Realistic part

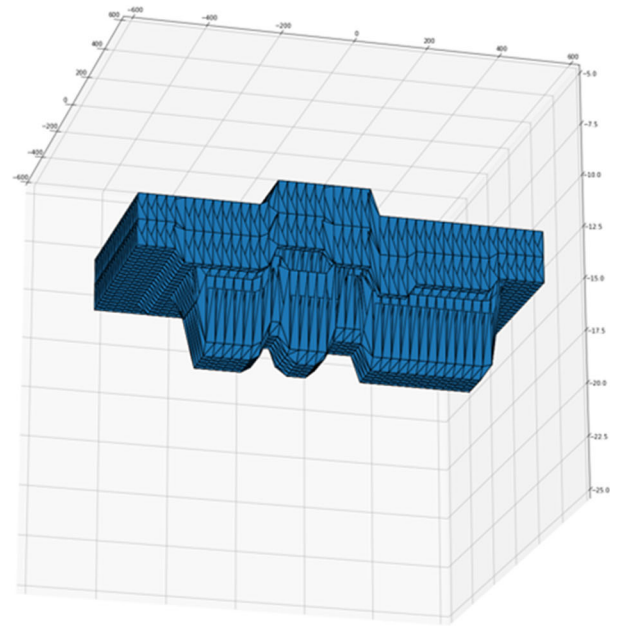
Our optimization reduced fill time by 32% , while ensuring complete fill of the part. On the realistic part we had an aerospace practitioner manually design a flow media distribution which we refer to hereafter as 'manually optimized'. The unoptimized, manually optimized, and RL optimized flow media distributions and fill progressions are displayed in [Figs. 19, 20 and 21](#).

## Discussion

On the Randomly Generated Test Part, the RL agent is successful at applying the knowledge gained during training to a new geometry that it has not seen before. In [Fig. 17](#), we can see that a flow media layout with 1 layer across the top of the preform suffers from racetracking causing trapoff. The instant optimization produced a flow media layout that eliminated trapoff. Inspecting the flow media layout, we can see that the agent has made two major changes to the default design with 1 layer of flow media across the top of the entire preform: adding a layer of flow media on the top of the outer pad-up to encourage fast resin flow to the inner pad-up, and removing all flow media near the outlet edge, to slow resin flow to the outlet and allow time for both pad-ups



Randomly Generated Part  
(Z-axis scale 60x for visualization)



Realistic Part  
(Z-axis scale 60x for visualization)

Fig. 16 Case study parts

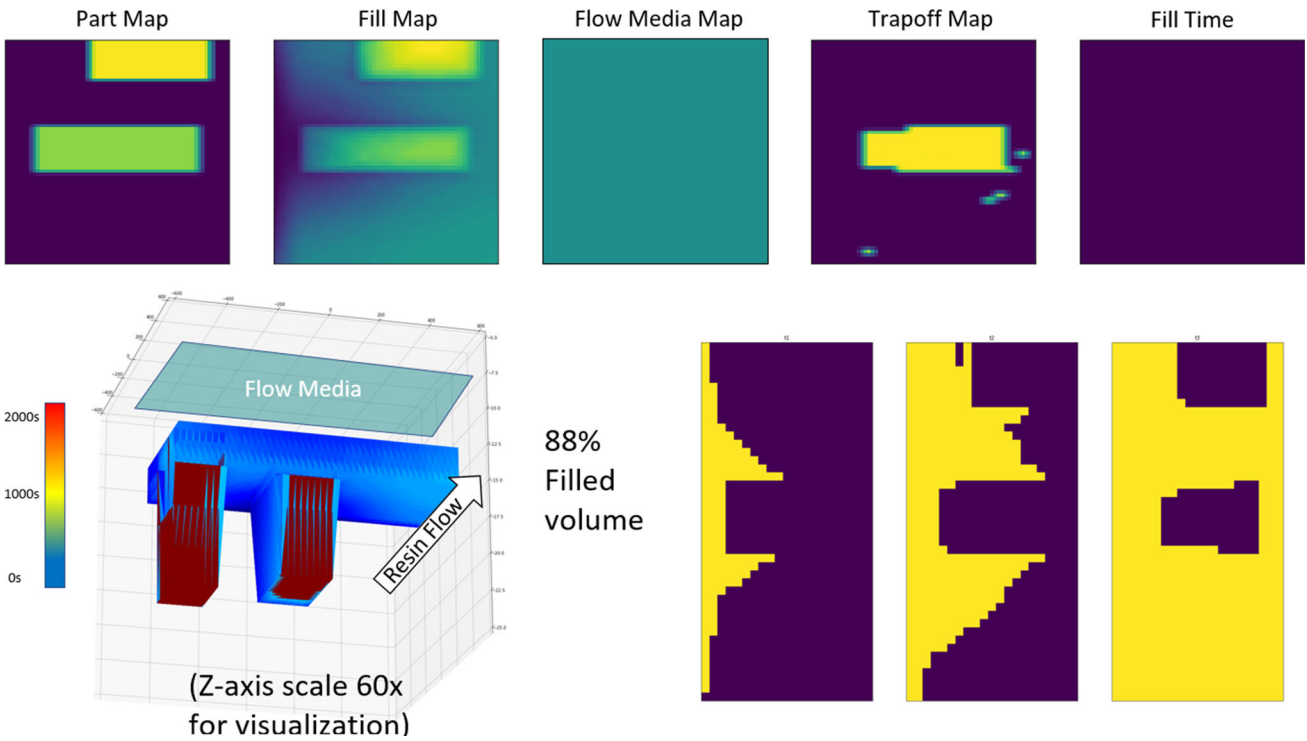


Fig. 17 Randomly generated test part pre-optimization



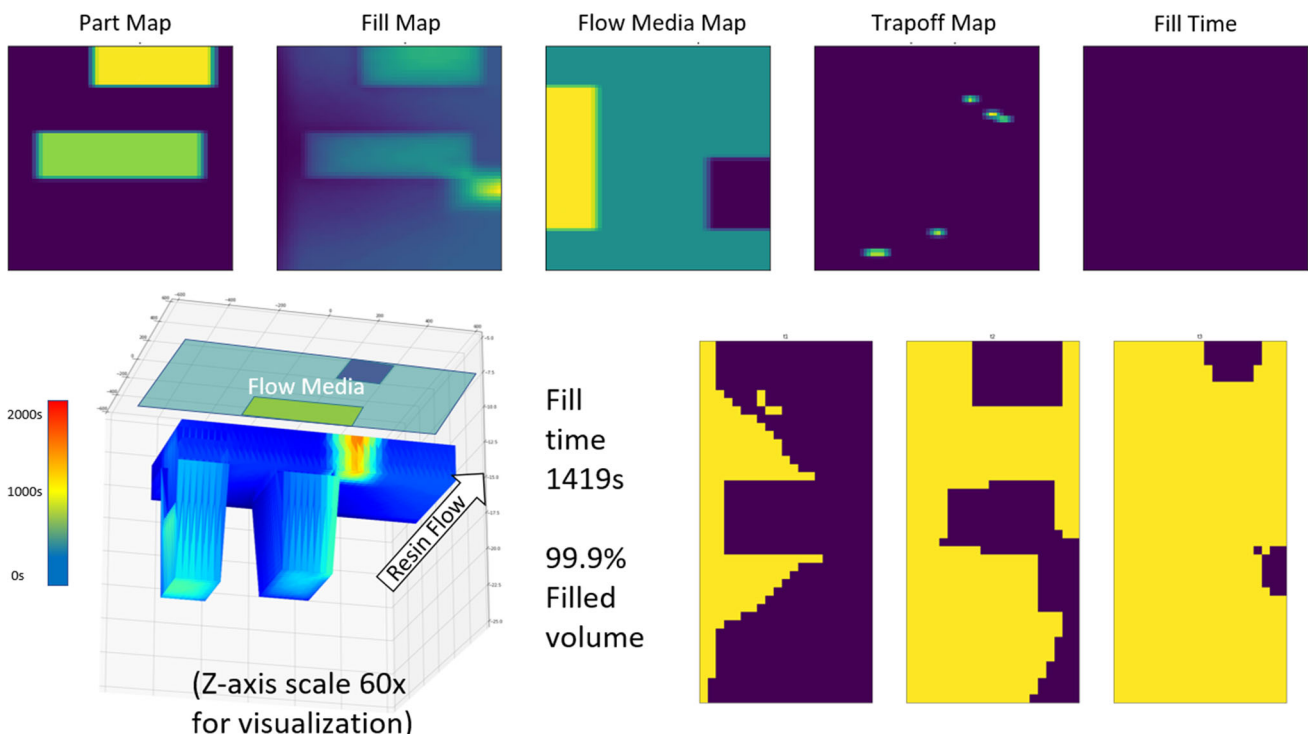


Fig. 18 Randomly generated test part post-optimization

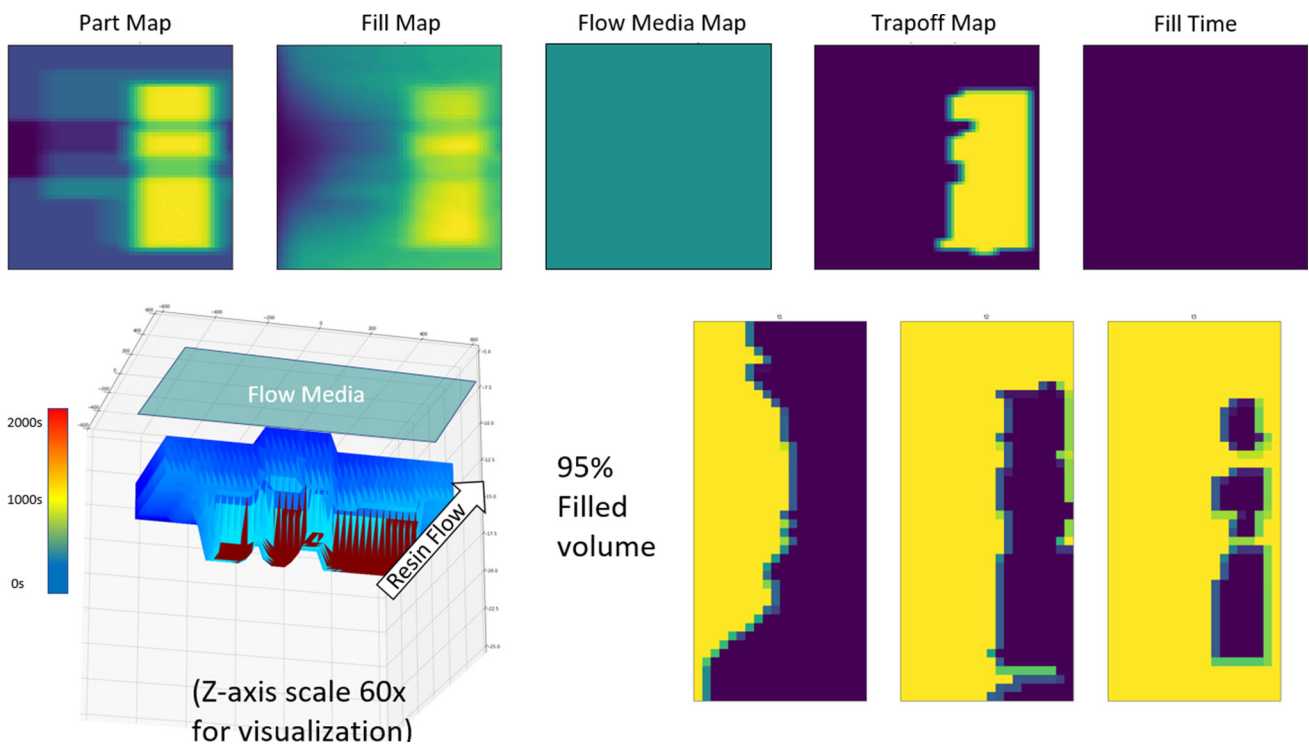


Fig. 19 Realistic part unoptimized

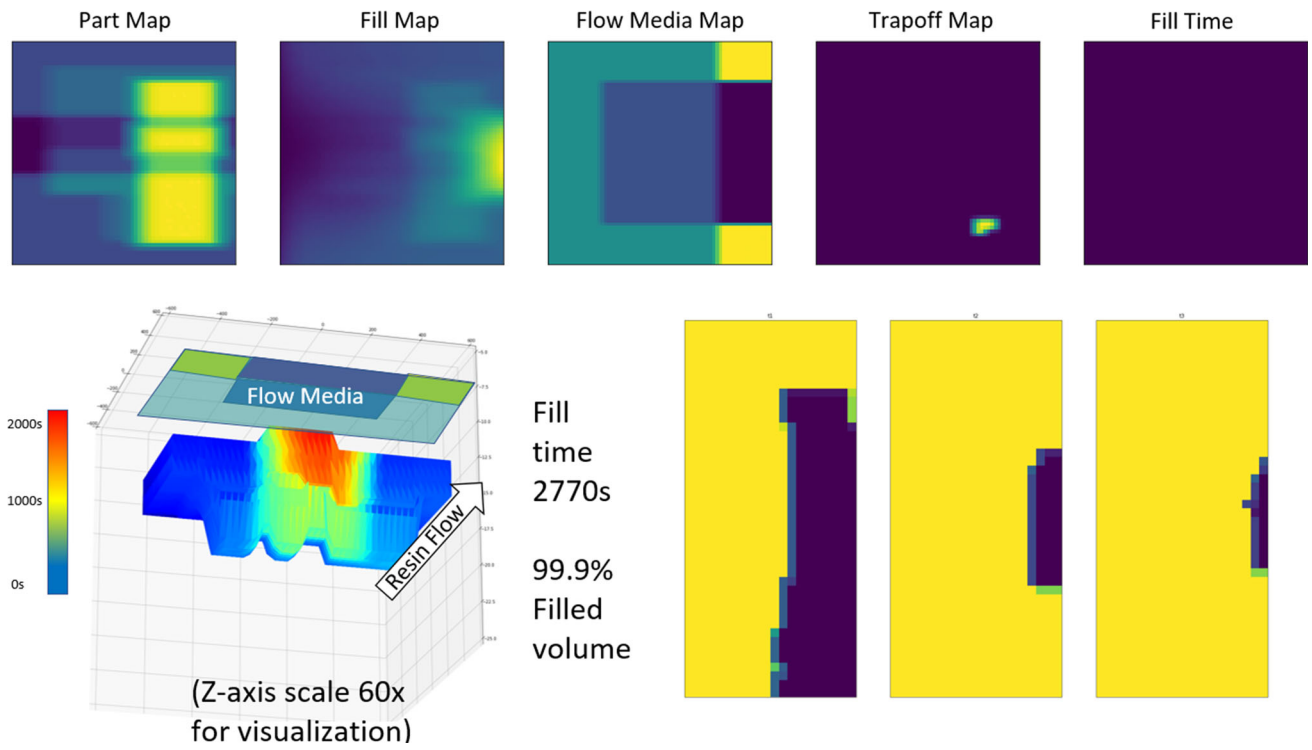


Fig. 20 Realistic part manually optimized to avoid trapoff

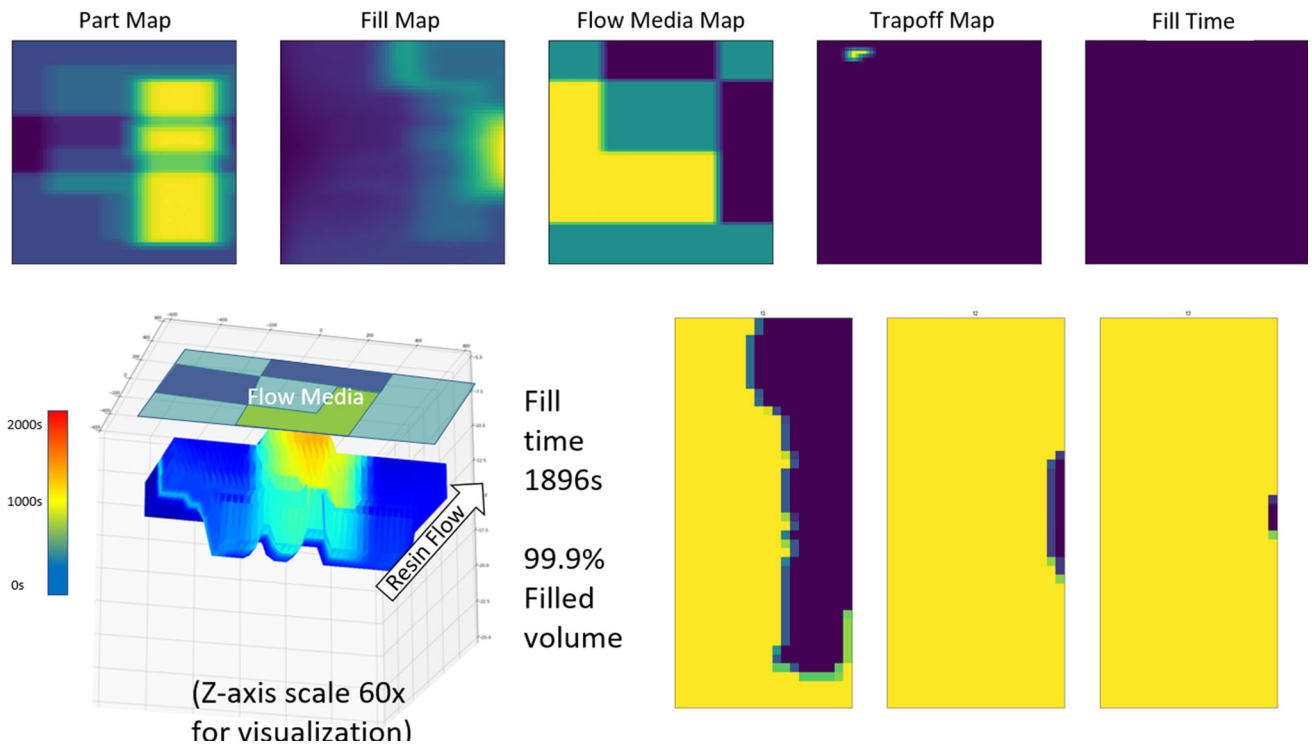


Fig. 21 Realistic part RL optimized

**Table 2** Efficiency comparison

Technique	Design Space	Evaluated Initially	Evaluated Per Optimization
GA (Hsiao et al., 2004)	2916	0%	16%
Tree Search (Sas et al., 2015)	16,384	0%	6.25%
<b>DRL (Ours)</b>	<b>531,441</b>	<b>3%</b>	<b>0.0001%</b>

to completely fill (see Figs. 17 and 18). This result agrees with intuition and is especially encouraging, as the agent has shown the ability to transfer knowledge from laminates seen during training, and apply it instantly to a new previously unseen geometry.

Applying the agent to a realistic geometry from the same thin laminate family instantly is the fundamental test of the method. Figs. 19, 20, and 21, show the unoptimized, manually optimized, and RL optimized flow media placements and flow progressions on the Realistic Part. In Fig. 19, we can see that a naive flow media placement again suffers from racetracking causing trapoff and ends in significant sections of the part remaining unfilled. The manually optimized placement (Fig. 20) has reduced flow media in the area between the flow runner and the thick area on the opposite edge of the part, retarding flow and avoiding racetracking along the top of the panel before the pad-up areas are filled through their thickness. This has the effect of eliminating the trapoff, however the fill time is significant for a part of this size. The RL flow media placement (Fig. 21) uses a similar tactic, reducing the flow to the pad-up regions by completely removing flow media on the edge of the part. It is apparent that there are two main differences between the RL and manual placements: asymmetry, and the number of layers used in less flow-critical sections of the part. By visual inspection we can see that the asymmetry introduced by the RL policy mirrors the asymmetry in the part thickness, as while the model ensures that the resin does not flow too quickly over the center-middle pad-up, it has found that offsetting a permeable path with 2 flow media layers across the top of the panel (the 'L'-shaped highway) allows a faster fill time overall. We also note the agent uses the same tactic to retain and increase flow media in the right-top and right-bottom corners where the panel is thin and will fill through-thickness easily. While not obviously parsimonious, the choice of 0 layers of flow media at the center-top position of the panel is technically a valid design, as vacuum is applied at all edges of the panel so this offset slow to fill area does not cause trapoff and still results in a complete fill. This tailoring of the flow distribution network to increase flow in less critical areas wherever possible, while retaining a focus on retarding flow over the thickest areas results in both trapoff avoidance and improvement in fill time, and shows the model has captured the dynamics of the process and how they relate to local part thickness and flow media density.

As all optimizations were done by our trained agent 'blind' (without running the process model at all), our case studies demonstrate that the trained agent is able to apply its learned process knowledge to realistic part geometries that were not seen during the training process and we have therefore have achieved our objective of knowledge transfer. On the Realistic Part, achieved improvement of 32% in fill time while maintaining complete fill when compared to an manually designed placement is industrially relevant. When applied to the Realistic Part and the Randomly Generated Test Part, our agent proposed the optimized flow media design in less than a second, demonstrating successful knowledge transfer. In both cases, the agent reached the highest reward configuration in one timestep, however this is not always the case (see Fig. 13 for an example of where the agent takes two steps to arrive at the highest reward configuration).

When evaluated on 100 new randomly sampled panels using the training mesh generation parameters the average entropy of the predicted multinomial action distribution was very close to zero (See Table 3), implying the agent is very confident in its actions. This is likely because the agent has overfit the training set and that the test and training sets are similar enough that meaningful generalisation has occurred. However, this overfitting is the goal in learning an optimal placement skill in a constrained scope. These results suggest that our approach could accommodate a more challenging and diverse set of geometries than used in this study. We leave this to future work.

Comparison to Hsiao et al. (2004) and Sas et al. (2015) is not straightforward due to differences in focus and approach, e.g. While Hsiao et. al. focus on optimization, Sas et. al. focus on robustness. However, both Hsiao and Sas propose direct optimization methods - i.e. they run the process model many times on the geometry of interest in a search procedure to explore the space of possible designs and arrive at a high-performing one. This process occurs for every new laminate geometry and given the process model takes significant time to run, the number of evaluated designs required to arrive at a good result (the search efficiency) is crucial to the performance of the overall method. In Table 2 we can see the relative search efficiencies of the methods as reported in the literature.

By contrast, our approach is completely different - we do not run the process model many times on the geometry of interest to explore the space. Instead, we initially evaluate a

**Table 3** Trained agent action entropy across 100 random panels

Mean	Std. Dev.
0.010	0.067

number of designs on a number of different geometries. This gives our trained agent an intuition about how to design, much like a person would have. Based on that knowledge gained upfront, our agent is able to propose a design *without running the process model at all*, instead transferring its learned knowledge to propose successful designs that lead to complete fill and reduced time to fill. During training, our method explores less space than a single run of Hasio or Sas, and during application on to similar geometries, it doesn't need to do any exploration at all - effectively achieving close to 100% efficiency on geometries in the family it was trained on.

## Limitations

We have demonstrated knowledge transfer within our domain of thin laminates with pad-ups for control surfaces. Within this constraint, the geometry can be varied while maintaining the instant optimization capability. Currently, changes to the flow runner location, gate location, material properties, or flow media grid (something other than 4x3) currently require retraining the agent. While material properties are unlikely to change during a design phase, a more dense flow media grid or a modified flow runner location are common enough needs that the technique should be extended in the future to be able to also handle these changes without retraining. While instant design assistance for a single geometry family is industrially relevant, for design assistance of this type to be generally useful, the agent would need to adapt to major changes in geometry such as more complex 3D integrated structure. We have not attempted to train an agent for that level of complexity. Further, evaluating robustness in the training by repeating the process with many random seeds, collecting training statistics across many runs, and tuning hyperparameters would be valuable in building confidence in the method.

## Conclusion

In this work, we introduced a novel instant optimization method for resin flow distribution networks based on learned knowledge transfer. We framed the problem flow media placement in terms of reinforcement learning, training an agent using a 3D Finite Element based process model of resin flow in carbon preforms. Trained on 10 panels from a thin laminate geometry family, our agent learns to place flow

media in order to avoid resin starvation (trapoff), and reduce total infusion time. Using a case study on a realistic aerospace part with a complex 12-dimensional flow media network, we show our method reduces fill time by 32% when compared to a manually designed placement, while maintaining the same fill quality. Due to the knowledge the agent has gained during training, it is able to optimize this previously unseen panel instantly (in less than a second). To our knowledge, this is the first example of knowledge transfer in this domain. Natural extensions of this work include studying the application of DRL-based optimization to very high dimensional liquid molding design spaces such as arbitrary flow runner pipe paths on complex 3D parts.

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s10845-022-01990-5>.

**Acknowledgements** The authors would like to thank Dr. Luke Fletcher for review.

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions

**Funding** This research was funded by Australian Government Commonwealth Supported Postgraduate Place under the Research Training Scheme. Support from Boeing was limited to data and model provision and accommodation of Study Leave.

**Availability of data and material** The code is subject to a confidentiality agreement and is not publicly available, a subset may be made available upon request, please contact the authors.

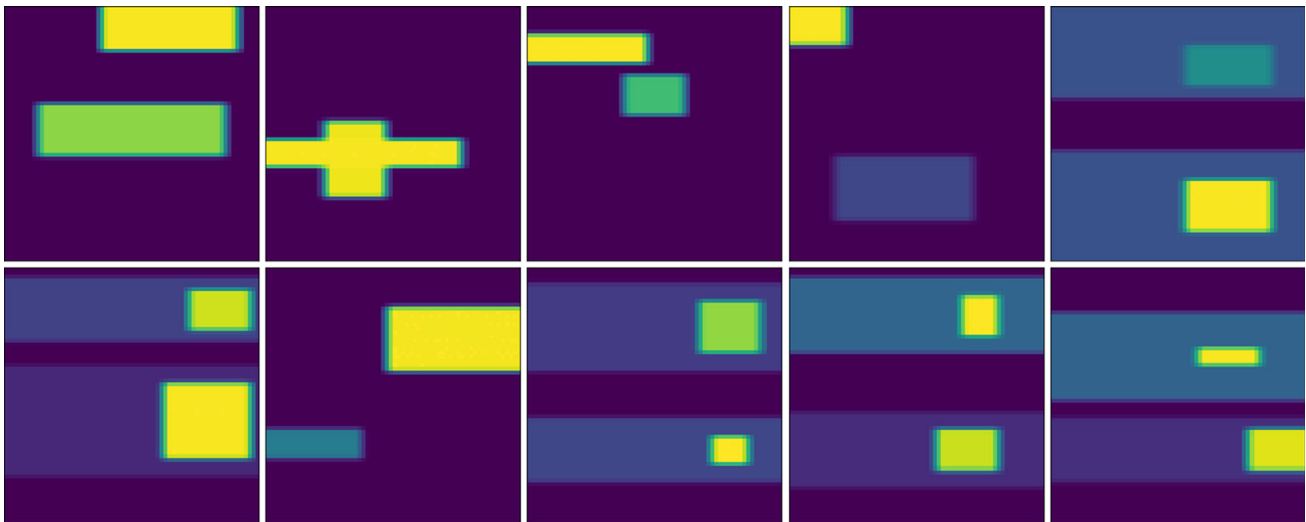
**Code Availability** The code is subject to a confidentiality agreement and is not publicly available, a subset may be made available upon request, please contact the authors.

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix A: Supplementary Material

See Fig. 22.



**Fig. 22** A random sample of 10 padups used in training

## References

- Andersen, R.E., Madsen, S., & Barlo, A.B., et al. (2019). Self-learning Processes in Smart Factories: Deep Reinforcement Learning for Process Control of Robot Brine Injection. *Procedia Manufacturing* 38:171–177. 29th International Conference on Flexible Automation and Intelligent Manufacturing ( FAIM 2019), June 24–28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing. <https://doi.org/10.1016/j.promfg.2020.01.023>, <https://www.sciencedirect.com/science/article/pii/S235197892030024X>.
- Boccard, A., Lee, W. I., & Springer, G. S. (1995). Model for determining the vent locations and the fill time of resin transfer molds. *Journal of Composite Materials*, 29(3), 306–333.
- Brockman, G., Cheung, V., & Pettersson L, et al. (2016). OpenAI Gym. [arXiv:1606.01540](https://arxiv.org/abs/1606.01540)
- Bruschke, M., & Advani, S. G. (1990). A finite element/control volume approach to mold filling in anisotropic porous media. *Polymer composites*, 11(6), 398–405.
- Dalcin, L. D., Paz, R. R., Kler, P. A., et al. (2011). Parallel distributed computing using Python. *Advances in Water Resources*, 34(9), 1124–1139.
- George, A. (2011). Optimization of resin infusion processing for composite materials: simulation and characterization strategies
- Gokce, A., Hsiao, K. T., & Advani, S. G. (2002). Branch and bound search to optimize injection gate locations in liquid composite molding processes. *Composites Part A: Applied Science and Manufacturing*, 33(9), 1263–1272.
- He, Z., Tran, K. P., Thomassey, S., et al. (2021). A deep reinforcement learning based multi-criteria decision support system for optimizing textile chemical process. *Computers in Industry*, 125(103), 373.
- Hiken, A. (2017). The evolution of the composite fuselage—a manufacturing perspective. *SAE International Journal of Aerospace* 10(2017-01-2154):77–91
- Hsiao, K.T., & Heider, D. (2012). 10 - Vacuum assisted resin transfer molding (VARTM) in polymer matrix composites. In: Advani SG, Hsiao KT (eds) *Manufacturing Techniques for Polymer Matrix Composites (PMCs)*. Woodhead Publishing Series in Composites Science and Engineering, Woodhead Publishing, p 310 – 347, <https://doi.org/10.1533/9780857096258.3.310>, <http://www.sciencedirect.com/science/article/pii/B9780857090676500109>
- Hsiao, K. T., Devillard, M., & Advani, S. G. (2004). Simulation based flow distribution network optimization for vacuum assisted resin transfer moulding process. *Modelling and Simulation in Materials Science and Engineering*, 12(3), S175.
- Kahng, A.B. (2021). Advancing placement. In: *Proceedings of the 2021 International Symposium on Physical Design*, pp 15–22
- Kessels, J. F., Jonker, A. S., & Akkerman, R. (2007). Optimising the flow pipe arrangement for resin infusion under flexible tooling. *Composites Part A: Applied Science and Manufacturing*, 38(9), 2076–2085.
- Lin, M., Murphy, M., & Hahn, H. (2000). Resin transfer molding process optimization. *Composites Part A: Applied Science and Manufacturing*, 31(4), 361–371.
- Ma, Y., Kassler, A., Ahmed, B.S., et al. (2022). Using Deep Reinforcement Learning for Zero Defect Smart Forging. [arXiv:2201.10268](https://arxiv.org/abs/2201.10268)
- Maier, R., Rohaly, T., Advani, S., et al. (1996). A fast numerical method for isothermal resin transfer mold filling. *International Journal for Numerical Methods in Engineering*, 39(8), 1405–1417.
- Mathur, R., Advani, S. G., & Fink, B. K. (1999). Use of genetic algorithms to optimize gate and vent locations for the resin transfer molding process. *Polymer composites*, 20(2), 167–178.
- Mirhoseini, A., Pham, H., Le, Q.V., et al. (2017). Device placement optimization with reinforcement learning. In: *International Conference on Machine Learning*, PMLR, pp 2430–2439
- Mirhoseini A, Goldie A, Yazgan M, et al. (2020). Chip Placement with Deep Reinforcement Learning. [arXiv preprint arXiv:2004.10746](https://arxiv.org/abs/2004.10746)
- Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529–533.
- Moriyama, T., De Magistris, G., Tsubori, M., et al. (2018). Reinforcement Learning Testbed for Power-Consumption Optimization. *Methods and Applications for Modeling and Simulation of Complex Systems* (pp. 45–59). Singapore: Springer Singapore.
- Panzer, M., & Bender, B. (2021). Deep reinforcement learning in production systems: a systematic literature review. *International Journal of Production Research* pp 1–26
- Raffin, A., Hill, A., & Ernestus, M., et al. (2019). Stable Baselines3. <https://github.com/DLR-RM/stable-baselines3>

- Roberts, T. (2007). Rapid growth forecast for carbon fibre market. *Reinforced Plastics*, 51(2), 10–13.
- Sánchez, F., Domenech, L., García, V., et al. (2015). Fast and reliable gate arrangement pre-design of resin infusion processes. *Composites Part A: Applied Science and Manufacturing*, 77, 285–292.
- Sas, H. S., Šimáček, P., & Advani, S. G. (2015). A methodology to reduce variability during vacuum infusion with optimized design of distribution media. *Composites Part A: Applied Science and Manufacturing*, 78, 223–233.
- Schulman, J., Wolski, F., & Dhariwal, P., et al. (2017). Proximal policy optimization algorithms. arXiv preprint [arXiv:1707.06347](https://arxiv.org/abs/1707.06347)
- Sigaud, O., & Buffet, O. (2013). *Markov decision processes in artificial intelligence*. John Wiley & Sons.
- Silver, D., Huang, A., Maddison, C. J., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587), 484–489.
- Soutis, C. (2005). Carbon fiber reinforced plastics in aircraft construction. *Materials Science and Engineering: A*, 412(1–2), 171–176.
- Sozer, E., Simacek, P., & Advani, S. (2012). 9 - Resin transfer molding (RTM) in polymer matrix composites. In: Advani SG, Hsiao KT (eds) *Manufacturing Techniques for Polymer Matrix Composites (PMCs)*. Woodhead Publishing Series in Composites Science and Engineering, Woodhead Publishing, p 245 – 309, <https://doi.org/10.1533/9780857096258.3.243>, <http://www.sciencedirect.com/science/article/pii/B9780857090676500092>
- Sutton, R.S., & Barto, A.G. (2018). *Reinforcement learning: An introduction*. MIT press
- Szarski, M., & Chauhan, S. (2021). Composite temperature profile and tooling optimization via Deep Reinforcement Learning. *Composites Part A: Applied Science and Manufacturing*, 142(106), 235.
- Vinyals, O., Babuschkin, I., & Chung, J., et al. (2019). AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>
- Wang, J., Simacek, P., & Advani, S. (2017). Use of medial axis to find optimal channel designs to reduce mold filling time in resin transfer molding. *Composites Part A: Applied Science and Manufacturing*, 95, 161–172.
- Young, W. B. (1994). Gate location optimization in liquid composite molding using genetic algorithms. *Journal of Composite Materials*, 28(12), 1098–1113.
- Yun, M., Carella, T., Simacek, P., et al. (2017). Stochastic modeling of through the thickness permeability variation in a fabric and its effect on void formation during Vacuum Assisted Resin Transfer Molding. *Composites Science and Technology*, 149, 100–107.
- Zimmerling, C., Poppe, C., Stein, O., et al. (2022). Optimisation of manufacturing process parameters for variable component geometries using reinforcement learning. *Materials & Design*, 214(110), 423.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.