Machine Learning-Based Prediction of Fabrication Constraints in Large-Scale Robotic 3D Printing

Przemyslaw Roj
Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA.
przemyslaw.work@ku.edu

Nasab Palastri
Department of Computer Science, University of Central Florida, Orlando, FL, USA.
nasab.work@ucf.edu

Gianluigi Toscani
Department of Computer Science, University of Houston, Houston, TX, USA.
gianluigitoscani@uh.edu

Elmeri Hartikainen
Department of Computer Science, University of New Hampshire, Durham, NH, USA.
elmeri.hartikainen@unh.edu

Abstract

Large-scale robotic 3D printing extends material extrusion and related additive processes beyond the build envelope of conventional gantry systems, but it also couples freeform geometry, multi-axis robot kinematics, deposition dynamics, material state, and structural stability in ways that make fabrication constraints difficult to evaluate during design and path planning. Deterministic computer-aided manufacturing checks are useful for reachability and collision avoidance, yet they often treat process and structural constraints separately and provide little calibrated information about the probability that a proposed print will remain feasible under real operating variability. This manuscript develops a constraint-aware machine-learning framework for predicting fabrication feasibility margins in large-scale robotic additive manufacturing. The framework represents a candidate fabrication plan through geometry, toolpath, robot-configuration, material, and environmental features; defines a vector of constraint responses with associated thresholds; and learns a probabilistic mapping from plan descriptors to constraint margins. The training objective combines heteroscedastic regression with a feasibility penalty, and prediction uncertainty is handled through Gaussian-process or ensemble estimates followed by distribution-free calibration. Constraint predictions are then converted into planning decisions through a risk-constrained decision rule that balances expected utility against the probability of violation. The analytical assessment shows why false-feasible errors, path-dependent label noise, and distribution shift are the principal technical risks, and it explains how calibrated uncertainty, interpretable feature attribution, and constraint-specific baselines can reduce those risks. The framework provides a disciplined route for integrating machine-learning prediction into large-scale robotic printing without replacing physical constraint models.

Keywords

robotic additive manufacturing; fabrication constraints; probabilistic machine learning; uncertainty calibration; conformal prediction; constraint-aware planning

1. Introduction

Additive manufacturing has moved from rapid prototyping toward production-oriented fabrication, and reviews of the field identify process qualification, geometric fidelity, and material consistency as persistent engineering challenges [1]. The diversity of extrusion, photopolymerization, powder-bed, and direct-energy deposition processes means that fabrication constraints are not reducible to a single geometric rule; they depend on material rheology, thermal history, machine architecture, and toolpath design [2]. Large-scale robotic 3D printing intensifies this coupling because the manipulator, deposition head, material supply, and support structure interact across a workspace that may be comparable to the printed object itself. Robotic platforms have been developed for multi-functional digital fabrication [4], and multi-axis motion strategies have been introduced to reduce support requirements and improve freeform production [5]. Material systems for three-dimensional printing of soft and functional matter further broaden the range of admissible process windows [3].

The central planning problem is that a design or toolpath can appear geometrically valid while remaining infeasible for a specific robot, material, and deposition sequence. A candidate path may lie within the nominal workspace but produce poor joint-torque margins, exceed collision clearance near a support frame, require an extrusion rate that destabilizes bead geometry, or create a thermal history that compromises interlayer bonding. Conversely, conservative process rules exclude feasible plans and reduce the productive use of large robotic systems. The practical need is therefore not merely a classifier of printable versus nonprintable objects, but a predictor of continuous constraint margins that can be queried before fabrication and updated as the plan changes.

Machine-learning methods are attractive because they can approximate expensive physical simulations and capture interactions among geometry, path, and process variables. Their use in additive manufacturing is already established for process monitoring, property prediction, and parameter optimization [20]. However, a direct regression model trained only on successful prints can confuse correlation with feasibility, and an uncalibrated classifier can produce confident false-feasible predictions. Large-scale robotic printing also creates a label problem: constraint violations are rare, expensive to observe, and often path-dependent. A model that predicts average quality is not sufficient for planning; the planner needs an estimate of the margin to each constraint and a calibrated statement of uncertainty around that margin.

This paper contributes a constraint-aware prediction framework for large-scale robotic 3D printing. The framework defines fabrication constraints as thresholded response functions, represents each candidate plan through a structured feature vector, and learns a probabilistic map from plan descriptors to constraint margins. It combines heteroscedastic regression with a feasibility penalty, treats Gaussian-process and ensemble uncertainty as inputs to calibration, and converts calibrated predictions into risk-constrained planning decisions. The analytical assessment focuses on failure mechanisms that are specific to robotic printing: false-feasible errors, path-dependent noise, constraint interaction, and shift between training and

deployment geometries. The aim is a technically defensible modeling framework that can be integrated with existing computer-aided manufacturing and robot-planning pipelines.

2. Fabrication Constraints in Large-Scale Robotic Additive Manufacturing

2.1 Constraint classes and feasibility margins

A large-scale robotic print is feasible only when several constraint families are satisfied simultaneously. Geometric constraints include minimum feature size, overhang angle, unsupported span, mesh self-intersection, and clearance around the deposited bead. Kinematic constraints include reachability, joint limits, singularities, velocity and acceleration limits, and collision between the manipulator, deposition head, material hose, and printed structure. Process constraints include extrusion rate, pressure stability, bead width, layer adhesion, and idle-time limits that affect curing or cooling. Structural constraints include self-weight deflection, stability of the partially printed object, and support requirements. General additive-manufacturing guidance emphasizes that these constraints are process-dependent rather than universal [6], and standard treatments of additive technologies show how machine architecture and material delivery shape the admissible build envelope [7].

Let x denote a candidate fabrication plan and let g_k(x) denote the predicted response of the k-th constraint, such as available collision clearance, normalized joint-torque margin, or bead-width error. Each response has a threshold τ_k that separates acceptable from unacceptable behavior. The feasible set is then defined as follows.

Here X is the space of admissible plan descriptors, K is the number of constraint families, and F is the set of plans whose constraint responses meet all thresholds. The formulation is deliberately general: a constraint may be a physical distance, a normalized utilization ratio, or a process-window margin. In practice, some constraints are upper-bounded rather than lower-bounded, such as maximum acceleration or maximum bead-width deviation; those are converted to margin form by reversing the sign or by defining g_k as the negative of the violation measure. The important modeling choice is that prediction targets are margins, not binary labels, because margins preserve ordering among feasible plans and expose proximity to failure.

2.2 Why prediction is difficult

The first difficulty is nonlinear interaction. A path segment that is kinematically admissible at low speed may become infeasible when extrusion rate and acceleration are increased, and a geometry that is stable with a dense support lattice may deflect when supports are removed. The second difficulty is path dependence. Constraint responses depend on deposition order, layer cooling, and the evolving robot configuration, so two plans with similar final geometry can have different feasibility. The third difficulty is label scarcity. Violations are costly and rare, and simulated labels may be biased by modeling assumptions. The fourth difficulty is distribution shift. A model trained on one family of geometries or one robot cell may not transfer to a new material, a larger workspace, or a different deposition head.

Several machine-learning model classes are relevant. Random forests provide robust tabular regression and feature interactions without extensive preprocessing [8]. Gaussian processes provide mean and variance estimates for smooth response surfaces and are useful when data are limited [9]. Bayesian approximations through dropout offer scalable uncertainty estimates for neural networks [10], while deep ensembles provide a practical

alternative that often captures model disagreement [11]. Conformal prediction can add distribution-free coverage guarantees around a point predictor [12]. Interpretability methods such as accumulated local effects and feature-attribution summaries help determine whether a model is relying on physically plausible variables [13], and general interpretable-modeling practice emphasizes transparent validation of black-box behavior [14]. Calibration is essential because a predicted probability of feasibility is meaningful only if it matches observed frequencies; modern neural networks can be miscalibrated even when their accuracy is high [15]. These methods are complementary rather than mutually exclusive: a Gaussian process or ensemble can supply uncertainty, conformal calibration can correct coverage, and interpretability can diagnose physically implausible learned relationships.

3. Methodological Framework for Constraint Prediction

3.1 Plan representation and prediction target

The framework begins with a structured representation of each candidate plan. Geometry features include mesh volume, surface-area-to-volume ratio, local overhang distribution, minimum feature thickness, support volume fraction, and spatial occupancy descriptors. Toolpath features include layer height, bead width, extrusion velocity, acceleration profile, path curvature, idle time, and the sequence of deposition segments. Robot features include joint configuration, manipulability, torque utilization, reach margin, and collision clearance along sampled waypoints. Material features include viscosity or curing state, temperature, and pot life where applicable. Environmental features include ambient temperature and humidity when they affect deposition.

The prediction target is a vector y of constraint responses, not a single printability score. A model f_θ with parameters θ maps the plan descriptor x to an estimate of the expected response vector.

$$\hat{y}(x) = f_\theta(x) \approx \mathrm{E}[y|x]$$

In this expression, ŷ is the predicted constraint vector, f_θ is the learned function, and E[y | x] is the conditional expectation of the response vector given the plan. The vector formulation allows the same model to share information across constraints while retaining constraint-specific thresholds. For planning, however, a single aggregate prediction is insufficient; the model must preserve the identity of each constraint so that a low clearance margin is not averaged away by a comfortable torque margin.

3.2 Feature construction and physical consistency

Feature construction should respect the physical meaning of the constraints. Collision clearance, for example, is better represented by signed distance samples along the toolpath than by a single global bounding-box distance. Joint-torque margin is better represented by normalized torque utilization along the most demanding trajectory segment than by an average over the whole path. Process stability is better represented by the rate of change of extrusion pressure and path velocity than by their nominal values alone. Sampling strategies should therefore produce features that are aligned with the mechanism that causes violation.

Geometric representation deserves particular attention in large-scale robotic printing because the mesh is not only a visual object but a constraint on support, collision, and deposition order. Mesh optimisation can alter the distribution of overhangs, unsupported spans, and self-intersections in spatial wireframe and lattice structures [16]. Such changes propagate into the feature vector and therefore into predicted feasibility. The model should

not be trained on a single mesh representation without recording the preprocessing decisions that produced it, because mesh repair, decimation, and support generation can shift the distribution of geometric features and create silent distribution shift.

A second consistency requirement is that features should be computed from quantities available before fabrication. If a feature requires the completed print or a destructive measurement, it cannot serve as a planning input. This constraint favors features derived from computer-aided design, robot kinematics, material specifications, and calibrated process models. The resulting feature set is necessarily incomplete, which is why uncertainty estimation is part of the framework rather than an optional post-processing step.

3.3 Probabilistic learning and uncertainty estimation

The model should output both a mean constraint response and a variance or ensemble spread. A practical training objective for a heteroscedastic regression model is a Gaussian negative log-likelihood with an added feasibility penalty. The penalty increases the loss when a predicted constraint falls below its threshold, even if the point prediction is close to the observed value.

$$L(\theta) = \sum_i \, (y_i - f_\theta(x_i))^2/(2\sigma_i^2) + ½ \, log \, \sigma_i^2 \; + \; \lambda \sum_k \, max(0, \tau_k - f_\{\theta, k\})^2$$

Here L is the training loss, θ are model parameters, y_i is the observed response for training plan i, f_θ(x_i) is the predicted mean, σ_i² is the predicted variance for that observation, λ controls the strength of the feasibility penalty, f_{θ,k} is the predicted response for constraint k, and τ_k is its threshold. The first two terms encourage accurate means and well-scaled variances; the final term imposes asymmetric cost on predictions that place a plan below a required margin. The asymmetry is intentional because false-feasible errors are usually more damaging than false-infeasible errors: the former can lead to a failed print, collision, or unsafe robot motion, whereas the latter may only cause an unnecessary redesign.

Gaussian-process regression provides a transparent baseline for low-dimensional or smooth constraint responses. Given training inputs and responses, the posterior mean and variance at a new plan x are expressed through a kernel function k and a noise variance σ_n².

$$\mu_*(x) = k_*{}^T (K + \sigma_n^2 I)^{-1} y$$

$$\sigma_*{}^2(x) = k(x, x) - k_*{}^T (K + \sigma_n^2 I)^{-1} k_*$$

In these equations, K is the kernel matrix among training inputs, k_ is the vector of kernel evaluations between the new input and the training inputs, I is the identity matrix, μ_(x) is the posterior mean, and σ_²(x) is the posterior variance. The Gaussian-process formulation is valuable when constraint data are sparse and the response surface is expected to vary smoothly with continuous path parameters. Deep ensembles can be used when the feature space is high-dimensional or contains categorical plan descriptors, with the ensemble variance serving as an empirical uncertainty estimate [11]. In either case, the uncertainty estimate should be validated against held-out constraint errors rather than assumed to be meaningful by construction.

3.4 Calibration and distribution-free coverage

A probabilistic model can be accurate on average and still be overconfident near threshold boundaries, which is precisely where planning decisions are made. Calibration should therefore be constraint-specific. A model may produce well-calibrated bead-width predictions but poorly calibrated collision-clearance predictions because the latter are dominated by rare

geometric configurations. Conformal prediction offers a distribution-free way to construct prediction intervals around a point predictor using a held-out calibration set [12]. A generic interval for a scalar constraint response can be written as follows.

$$C_\alpha(x) = \{y : |y - f(x)| \ \leq \ q_\{1 - \alpha\}\}$$

Here C_α(x) is the calibrated prediction set at coverage level 1−α, f(x) is the point prediction, and q_{1−α} is the empirical quantile of absolute residuals on the calibration set. The construction assumes exchangeability between calibration and deployment plans, an assumption that must be examined when geometries, materials, or robot cells change. The broader theory of conformal prediction formalizes this logic for arbitrary predictors and nonparametric uncertainty [17]. In practice, constraint-specific conformal calibration can be applied separately to each margin, or jointly through a Bonferroni-style adjustment when simultaneous coverage across constraints is required.

Training a probabilistic model also requires optimization choices that affect reproducibility. Stochastic gradient methods such as Adam are common for neural models [18], but the optimizer is secondary to the data and feature design. The more consequential choices are the train-validation split, the stratification of plans by geometry family and robot configuration, and the treatment of near-threshold cases. If the calibration set contains only comfortable feasible plans, the conformal quantile will be too small for the boundary region where the planner actually operates. Calibration should therefore be enriched with plans that are near constraint thresholds, generated through controlled variation of path parameters rather than through unrepresentative random sampling.

3.5 From prediction to planning decision

The predictive model is useful only if its outputs can be converted into a planning decision. Let a denote a candidate action, such as selecting a toolpath, changing the robot base placement, adding a support, or reducing deposition speed. The planner should maximize expected utility subject to a calibrated limit on the probability of constraint violation.

$$a* \ = \ argmax_a \, E[U(a, y)] \, subject \, to \, P(min_k \, m_k \ \geq \ 0) \ \geq \ 1 - \alpha$$

In this expression, a is the selected action, U(a,y) is the utility of action a under the realized constraint response y, m_k is the margin for constraint k, and 1−α is the required feasibility probability. The minimum margin captures the weakest constraint, but the probability statement should be computed from the calibrated joint predictive distribution rather than from independent marginal intervals. If the joint distribution is unavailable, a conservative union bound can be used, or the planner can optimize a risk-sensitive objective that penalizes low margins across all constraints. The decision rule makes the trade-off between productivity and risk explicit: a higher required feasibility probability shrinks the admissible action set and may require slower printing, additional support, or a different robot configuration.

4. Analytical Assessment and Technical Discussion

4.1 Constraint interactions and failure modes

The most important failure mode is the false-feasible prediction. It occurs when the model predicts every constraint margin as nonnegative, but at least one physical constraint is violated during fabrication. False-feasible errors are dangerous because they can produce collisions, unstable prints, or material failure, and they are likely near the boundary of the feasible set where training data are sparse. The feasibility penalty in the training objective

reduces this risk, but it cannot eliminate it when the model is extrapolating. The planner should therefore use calibrated upper bounds on violation probability rather than point estimates alone.

A second failure mode is path-dependent label noise. Two plans with similar static geometry can have different feasibility because deposition order changes thermal history and robot configuration. If the model is trained on plan-level labels without path-sequence features, it will attribute the difference to geometry and learn a biased response surface. The feature set must include sequence-sensitive descriptors such as maximum idle time, layer completion time, and the order of overhang segments. A third failure mode is constraint masking: a model may appear accurate on average because most constraints are comfortably satisfied, while its errors concentrate on the one constraint that actually limits the print. Constraint-wise evaluation is therefore necessary, and aggregate accuracy is misleading.

Process monitoring and control provide a complementary source of evidence [19]. A prediction model estimates feasibility before fabrication, while in-process sensing can detect deviation from the predicted regime. The two should be linked: monitoring data can update the model's uncertainty, and the planner can define fallback actions when the observed process state leaves the calibrated envelope. This link is particularly relevant for large-scale robotic printing, where a small deviation early in a long path can accumulate into a significant geometric or structural error.

4.2 Baselines and model-selection logic

A credible evaluation must compare against deterministic baselines. The simplest baseline is a kinematic and geometric checker that evaluates reachability, joint limits, and collision clearance along sampled waypoints. A second baseline is a physics-based simulation or reduced-order model of extrusion and structural response. These baselines are not obsolete; they encode mechanisms that a purely data-driven model may fail to learn. The machine-learning model should be judged by whether it adds calibrated predictive information beyond them, especially near threshold boundaries and for plan families that are expensive to simulate.

Random forests are a strong baseline for tabular features and small data sets [8], while Gaussian processes are preferable when uncertainty must be smooth and interpretable [9]. Deep ensembles and Bayesian dropout are appropriate when the feature space is large or includes high-dimensional geometry descriptors [10,11]. Conformal calibration should be applied to the selected model, not treated as a separate model class. Model selection should use constraint-specific metrics: mean absolute error for margin prediction, false-feasible rate at a fixed feasibility probability, interval coverage, interval sharpness, and decision regret relative to a conservative planner. These metrics reflect the actual use of the predictor and prevent a model with low average error but poor threshold behavior from being selected.

4.3 Uncertainty, calibration, and operational thresholds

Uncertainty is not a single quantity. Aleatoric uncertainty arises from material variability, sensor noise, and small process fluctuations; epistemic uncertainty arises from limited coverage of the plan space. The planner should respond differently to each. High aleatoric uncertainty may justify a slower process or a more forgiving deposition strategy, while high epistemic uncertainty should trigger additional simulation, a conservative fallback, or a request for more data in that region of the plan space. Calibration makes this distinction actionable because a calibrated interval can be translated into a probability of violation at the chosen threshold.

The operational threshold $1-\alpha$ is a design parameter, not a statistical afterthought. A high threshold reduces the risk of false-feasible errors but increases the rate at which feasible plans are rejected. The appropriate value depends on the cost of failure, the cost of redesign, and the safety requirements of the robot cell. Because the decision rule in Equation 7 uses the calibrated joint distribution, the threshold can be set explicitly by engineering policy and audited against observed outcomes. This is a stronger basis for qualification than an uncalibrated classifier score.

4.4 Computational cost and deployment

The computational cost of the framework has three parts: feature extraction, model inference, and calibration. Feature extraction is dominated by geometric sampling and robot-kinematics evaluation along the toolpath. Model inference is typically negligible compared with simulation, but the cost of uncertainty estimation can grow with ensemble size or the number of Gaussian-process inducing points. Calibration adds a held-out pass but does not affect online inference once the conformal quantile is fixed.

Deployment should be staged. Offline, the model can screen large design spaces and identify regions where constraint margins are low. Online, it can provide fast feasibility estimates during interactive path editing and robot-cell configuration. The model should not be used as a replacement for final collision checking or safety certification; rather, it should rank candidates and expose risk so that deterministic checks can be focused where they matter. This division of labor preserves the strengths of physics-based verification while using machine learning to reduce the number of expensive evaluations.

5. Conclusion

Large-scale robotic 3D printing requires fabrication decisions that respect geometric, kinematic, process, and structural constraints simultaneously. This paper has developed a constraint-aware machine-learning framework for predicting those constraints as continuous margins rather than binary labels. The framework represents each plan through geometry, toolpath, robot, material, and environmental features; trains a probabilistic model with a feasibility penalty; calibrates uncertainty with conformal methods; and converts calibrated predictions into risk-constrained planning decisions. The analytical assessment identifies false-feasible errors, path-dependent label noise, constraint masking, and distribution shift as the principal technical risks. It also shows why constraint-specific evaluation, interpretable feature attribution, and deterministic baselines are necessary for credible deployment. The central insight is that machine learning is most useful in this domain when it predicts margins with calibrated uncertainty and when its outputs are embedded in an explicit decision rule that respects fabrication risk. Such a framework can support faster exploration of large robotic build envelopes while keeping physical constraint checking, process monitoring, and engineering accountability at the center of fabrication planning.

References

1. Gao, W., Zhang, Y., Ramanujan, D., Ramani, K., Chen, Y., Williams, C. B., Wang, C. C. L., Shin, Y. C., Zhang, S., & Zavattieri, P. D. (2015). The status, challenges, and future of additive manufacturing in engineering. Computer-Aided Design, 69, 65–89. https://doi.org/10.1016/j.cad.2015.04.001

2. Ngo, T. D., Kashani, A., Imbalzano, G., Nguyen, K. T. Q., & Hui, D. (2018). Additive manufacturing (3D printing): A review of materials, methods, applications and

challenges. Composites Part B: Engineering, 143, 172–196. https://doi.org/10.1016/j.compositesb.2018.02.012

3. Truby, R. L., & Lewis, J. A. (2016). Printing soft matter in three dimensions. Nature, 540(7633), 371–378. https://doi.org/10.1038/nature21003

4. Keating, S. J., & Oxman, N. (2013). Compound fabrication: A multi-functional robotic platform for digital design and fabrication. Robotics and Computer-Integrated Manufacturing, 29(6), 439–448. https://doi.org/10.1016/j.rcim.2013.05.001

5. Dai, C., Wang, C. C. L., Wu, C., Lefebvre, S., Fang, G., & Liu, Y.-J. (2018). Support-free volume printing by multi-axis motion. ACM Transactions on Graphics, 37(4), Article 134. https://doi.org/10.1145/3197517.3201342

6. Huang, Y., Leu, M. C., Mazumder, J., & Donmez, A. (2015). Additive manufacturing: Current state, future potential, gaps and needs, and recommendations. Journal of Manufacturing Science and Engineering, 137(1), 014001. https://doi.org/10.1115/1.4028725

7. Gibson, I., Rosen, D., Stucker, B., & Khorasani, M. (2021). Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing (3rd ed.). Springer. https://doi.org/10.1007/978-3-030-56127-7

8. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

9. Rasmussen, C. E., & Williams, C. K. I. (2006). Gaussian processes for machine learning. MIT Press.

10. Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the 33rd International Conference on Machine Learning (pp. 1050–1059). PMLR.

11. Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in Neural Information Processing Systems 30 (pp. 6402–6413).

12. Angelopoulos, A. N., & Bates, S. (2021). A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv:2107.07511.

13. Apley, D. W., & Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 82(4), 1059–1086. https://doi.org/10.1111/rssb.12377

14. Molnar, C. (2022). Interpretable machine learning: A guide for making black box models explainable (2nd ed.). Independently published.

15. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning (pp. 1321–1330). PMLR.

16. Yu, L., & Luo, D. (2018). Mesh optimisation for spatial wireframe robotic 3d printing. International Journal of Rapid Manufacturing, 7(2-3), 169-185.

17. Vovk, V., Gammerman, A., & Shafer, G. (2005). Algorithmic learning in a random world. Springer.

18. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations.

19. Tapia, G., & Elwany, A. (2014). A review on process monitoring and control in metal-based additive manufacturing. Journal of Manufacturing Science and Engineering, 136(6), 060801. https://doi.org/10.1115/1.4028540

20. Qi, X., Chen, G., Li, Y., Cheng, X., & Li, C. (2019). Applying neural-network-based machine learning to additive manufacturing: Current applications, challenges, and future perspectives. Engineering, 5(4), 721–729. https://doi.org/10.1016/j.eng.2019.04.012