PFAB: A Generic Software Framework for Predictive Fabrication in Construction

Luca Bettermann[1,2], **Sebastian Esser**[1,2], **Martin Slepicka**[1,2] **and André Borrmann**[1,2]

[1]Chair of Computing in Civil and Building Engineering, Technical University of Munich, Germany
[2]Georg Nemetschek Institute, Technical University of Munich, Germany

luca.bettermann@tum.de, sebastian.esser@tum.de, martin.slepicka@tum.de, andre.borrmann@tum.de

Abstract -

As digital fabrication systems in construction become more sensorized and controllable, they expose growing opportunities for data-driven process understanding that remain largely underutilized. Consequently, additive manufacturing workflows continue to rely on expert intuition and trial-and-error, which limits their industrial adoption in dynamic environments, such as construction sites. This paper presents PFAB, an open-source Python-based framework that enables predictive fabrication workflows by structuring data, coordinating learning tasks, and guiding parameter selection. PFAB formalizes fabrication as a data-driven process that generates predictive models through active learning methods and supports performance-based optimization of process parameters. Its modular, schema-first design separates framework-level orchestration from fabrication-specific logic, allowing the same predictive workflow to be applied across fabrication systems. The framework is demonstrated through Learning–by–Printing, an exemplary implementation for extrusion-based additive manufacturing, and is complemented by an executable reference implementation illustrating end-to-end usage.

Keywords -

Predictive fabrication; Data-driven additive manufacturing; Active learning; Learning by printing

1 Introduction

1.1 Motivation

Additive Manufacturing (AM) for construction is widely regarded as a promising technology to address long-standing challenges in productivity and sustainability. By connecting parametric design with automated fabrication and machine control, AM offers unprecedented flexibility and enables the efficient fabrication of bespoke components for reduced material use [1]. Yet despite rapid growth in academic activity and high-visibility demonstrators, AM has not realized this potential in practice [2]. Most systems continue to depend on expert intuition and iterative trial-and-error, resulting in a lack of first-time-right manufacturing and limiting broader industrial adoption.

A central difficulty emerges from the interaction of multiple interdependent factors, including design parameters, material behavior, environmental conditions, and machine settings. Small variations in any of these aspects can produce disproportionately different results [3]. This coupling and variability make AM processes difficult to model explicitly and challenging to control using traditional methods. As a result, there is a growing need for approaches that can leverage the digital nature of AM not only for automation, but also for acquiring, structuring, and interpreting process data to enable more predictable and robust fabrication [4].

1.2 A layered view on digital fabrication

Digital fabrication workflows can be understood as a sequence of tightly connected layers. At the design layer, geometric intent, structural considerations, and material selection are defined using computational and parametric tools. The fabrication layer translates this intent into a task plan governed by process parameters and converts it into machine-executable instructions for physical execution. These two layers form the basis of current practice and are well supported by existing digital toolchains.

Modern digital fabrication systems increasingly operate as cyber-physical systems (CPS): sensors capture the *as-fabricated* reality, data feedback is used to synchronize fabrication tasks, monitor quality, and maintain stable operation [5]. However, the use of this data remains largely reactive. Data is collected, inspected, or used for post-process corrections, but rarely interpreted in a way that reveals the underlying relationships between the design, material, process, and environmental factors that shape fabrication outcomes. As a result, current workflows still lack a systematic mechanism for learning from data, and the potential of cyber-physical systems remains underutilized.

To fully exploit the data feedback provided by modern CPS architectures, a predictive layer is proposed (Fig. 1). The Predictive Fabrication (PFAB) framework introduces this layer through a modular and generalizable architecture that orchestrates the entire data-processing and learning pipeline. PFAB models the general behavior of a fabri-

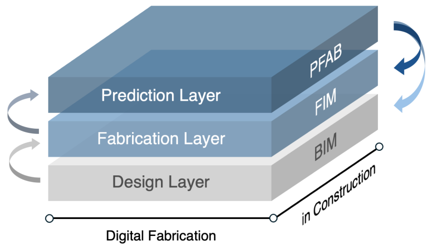

Figure 1. Layered view of digital fabrication in construction.

cation system to enable performance-driven optimization of process parameters. The resulting prediction model can be integrated into the fabrication layer for experiment-specific refinement and into the design layer for outcome-informed decision-making. This perspective provides the conceptual foundation for the generic software framework presented in this work.

2 Related Works

Recent work in digital fabrication highlights the strong coupling between design geometry, material behavior and process parameters, underscoring the need for structured process descriptions [6]. Frameworks such as Fabrication Information Modeling (FIM) formalize the digital-physical data flows required for cyber-physical operation [5]. Building on this foundation, data-driven approaches in AM have focused on feature extraction and learning methods to interpret sensing data and enable predictive modeling [4].

2.1 Data-driven additive manufacturing

Additive manufacturing processes build components layer by layer through controlled material deposition, giving rise to sensor measurements that are inherently spatially and temporally structured. Feature extraction forms a foundational component of data-driven workflows, as the performance of downstream learning processes is ultimately bounded by the quality and relevance of the extracted features. Wolfs et al. synthesized key quality indicators across scales and emphasized the importance of in-line and on-line measurement for achieving first-time-right manufacturing [7]. Versteege et al. demonstrated how knowledge-driven feature engineering can transform high-dimensional sensor data into compact, physically meaningful representations suitable for monitoring and learning [8, 9].

Complementary efforts use machine learning across a range of AM technologies to predict specific outcomes, such as buildability constraints [10] or microcrack investigations for the optimization of material performance [11]. Physics-informed and hybrid modeling strategies demonstrate how physical principles can be embedded into learning models, as shown in co-driven approaches that incorporate physical equations to improve rheological prediction and parameter selection [12]. Zhao et al. highlight the need for real-time monitoring to capture process deviations in-situ, as demonstrated by transformer-based detectors for in-process defect identification and analysis [13].

2.1.1 ML-oriented frameworks in related domains

Other fields have recognized the need for modular learning workflows that structure data processing, model training and deployment. In computational design, AIXD provides a low-code environment that integrates surrogate modeling and inverse design within an orchestrated pipeline [14]. General AutoML frameworks such as TPOT and Auto-Sklearn 2.0 [15, 16] further exemplify this principle by automating feature preprocessing, model selection, and validation within standardized pipelines. These frameworks demonstrate how modular architectures coordinate data, models and decisions, lowering the barrier to deploying data-driven techniques in complex learning workflows.

2.2 Summary of related works

Existing research shows that robust feature extraction remains fundamental, as predictive accuracy depends critically on the information derived from process measurements. However, current sensing, prediction, and hybrid modeling efforts address specific components of the fabrication process, leaving substantial potential in unifying these contributions within a holistic framework. More recently, studies have pointed to the need for real-time monitoring and adaptation during fabrication, where deviations must be addressed as they emerge to enable first-time-right manufacturing. At the same time, most approaches remain tightly coupled to specific materials, machines, or sensing setups, limiting their applicability across the diverse and variable characteristics of digital fabrication methods.

3 PFAB Framework

PFAB is a generic software framework for predictive fabrication that unifies data structuring, learning methods, and optimization routines within digital fabrication workflows. This transforms fabrication experiments into a systematic learning process, supporting parameter calibration and establishing the architecture for predictive closed-loop control. The PFAB framework is released as open-source software under the MIT license and cited as a versioned research artifact [17].

3.1 Overview

General-purpose machine learning frameworks assume the availability of static datasets and do not address a defining characteristic of fabrication systems: the ability to actively generate high-quality data through controlled physical experiments. PFAB is built around this capability, emphasizing targeted experimentation to maximize information acquisition while limiting resource expenditure.

The `PfabAgent` initiates this process by orchestrating the fabrication system through a structured data acquisition and evaluation workflow across an arbitrary number of sensing modalities. The resulting dataset enables the training of predictive models that support holistic, data-driven optimization of process parameters for unseen experimental configurations. The framework is built upon three architectural pillars:

- **Modular system architecture:** PFAB distinguishes between a core layer that defines the data object structure, a logic layer hosting fabrication-specific and internal processing, and an orchestration layer that integrates these components into a unified workflow.
- **Schema-first design:** To manage heterogeneous data, PFAB enforces a strict schema contract that guarantees consistency and enables automated validation across all downstream learning tasks.
- **Stateless agent:** The logic and orchestration are encapsulated in a stateless agent that operates as a pure function of the dataset, ensuring reproducibility and allowing robust recovery from interruptions.

Building on this architecture, PFAB structures parameter selection across three calibration modes: a *baseline mode* that proposes initial experiments through maximin spacing, an *exploration mode* focused on systematic information gain, and an *inference mode* that leverages trained models for performance-driven parameter optimization.

3.2 System requirements

PFAB assumes a digitally controlled fabrication system in which design intent, process parameters, and machine instructions are explicitly parameterized and programmable. Parameter configurations proposed by the predictive layer must be executable without manual intervention. The system must further provide sensor-based feedback capturing *as-fabricated* process information. This may include visual sensing, force or torque measurements, energy consumption, acoustic signals, or laser-based measurements. No specific sensing modality is required, as PFAB operates on derived features rather than raw signals.

3.3 Predictive layer workflow

The PFAB workflow consists of a sequence of stages that connect data structuring, user-defined logic, model training, and parameter calibration. While these steps can vary depending on the fabrication setup, their core structure follows a consistent pattern that integrates fabrication logic with PFAB's learning and calibration mechanisms. The key stages are outlined below.

3.3.1 Problem formalization

PFAB follows a schema-first architecture inspired by AIXD [14], in which all dimensions, parameters, features and performance attributes are defined explicitly before any data is collected. These data objects form a hierarchical structure: `Dimensions` describe the component structure, `Parameters` describe all inputs of the fabrication system, `Features` represent processed sensory information, and `PerformanceAttributes` score the extracted features to evaluate experiment outcomes. `DataBlocks` group these objects into coherent categories, and the resulting `Schema` serves as a strict contract that governs validation, storage and all downstream learning tasks. Once the schema is defined, a `Dataset` instance acts as a lightweight container that stores structured `ExpData` objects, one per fabrication experiment, and automatically manages their storage across local and external data sources (Listing 1).

```
# Definition of Dimensional Domain
dim1 = Dimension("layers", iterator="l_idx")
dim2 = Dimension("segments", iterator="s_idx")
d1   = Domain("spatial", [dim1, dim2])

# Definition of Data Objects
p1 = Parameter.real("speed", max_val=80.0)
f1 = Feature.array("filament_width", domain=d1)
a1 = PerformanceAttribute.score("print_quality")

# Create Data Blocks
params  = Parameters([p1])
domains = Domains([d1])
feats   = Features([f1])
attrs   = PerformanceAttributes([a1])

# Define Schema and initialize Dataset objects
schema = Schema("name", params, domains, feats, attrs)
dataset = Dataset(schema, local_folder="local/")
```

Listing 1. Initialization of schema and dataset.

Fabrication-specific logic is introduced by defining how raw sensor measurements are transformed into structured features and how these features are scored against task-specific target values to evaluate fabrication outcomes. Since the appropriate learning algorithm depends on the structure of the data, PFAB allows users to configure an appropriate prediction model. Once these components are specified and validated against the schema, the `PfabAgent` initializes the systems that orchestrate all subsequent evaluation, learning, and parameter calibration steps (Listing 2).

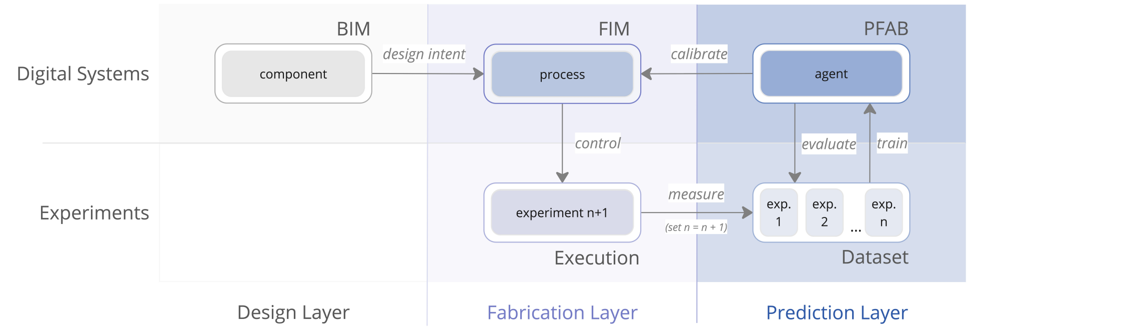

Figure 2. Illustration of the fabrication–learning loop with respect to the design, fabrication and prediction layers.

```
# Initialize PfabAgent
agent = PfabAgent()

# Import and register user-defined models
import FeatModel, EvalModel, PredModel
agent.register_model(FeatModel)
agent.register_model(EvalModel)
agent.register_model(PredModel)

# Initialize agentic systems and validate against schema
agent.initialize_systems(schema)
```

Listing 2. Agent initialization and user-defined model integration.

3.3.2 Calibration modes

PFAB structures parameter proposal across three named calibration modes that differ only in their objective function, all executed through a single optimization engine, the `CalibrationSystem`. This allows the framework to guide fabrication from the first experiment until production-ready deployment within a unified interface.

Mode	Prediction model	Objective
Baseline	-	Maximin spacing
Exploration	Surrogate	Information-driven
Inference	Trained	Performance-driven

Table 1. Modes of the `CalibrationSystem`.

Baseline mode. Before active model development can start, PFAB generates an initial dataset using the `PfabAgent.baseline_step`, which proposes experiment configurations by maximizing spacing within the defined parameter bounds, requiring no predictive model. The fabricated experiments form an initial schema-based `Dataset` that enables subsequent calibration steps to operate on empirical observations rather than priors.

Exploration mode. The structured dataset is interfaced with the `DataModule`, which handles normalization, batching, and dataset splitting according to the schema and user inputs, ensuring consistent preprocessing across all learning steps. In the exploration mode, the user invokes `PfabAgent.exploration_step` on the current datamodule, and PFAB proposes the most informative next parameter configuration within the defined bounds. The proposed experiment is fabricated and added to the dataset, incrementally improving data coverage and predictive fidelity. This closes the fabrication–learning loop illustrated in Fig. 2, yielding a progressively enriched dataset promoting predictive generalization. The prediction model is periodically trained and validated on the growing dataset using the datamodule via `PfabAgent.train_step`. Once sufficient data coverage and validation accuracy are achieved, the predictive layer is considered production-ready.

Inference mode. The trained prediction model is queried through `PfabAgent.inference_step` to map a given design intent, expressed through its parametric representation, to an optimized set of process parameters for fabrication. Unlike exploration, inference is guided by a performance-driven optimization objective rather than information acquisition. The objective is defined through weighted performance attributes, which encode task-specific trade-offs and are passed directly to the inference step, preserving the stateless nature of the agent while allowing objectives to be adjusted without corrupting the workflow.

3.4 Executable Reference Implementation

An executable reference repository is provided as the primary entry point to the framework. It instantiates the required fabrication-specific interfaces and executes the full agentic workflow using synthetic data, enabling end-to-end execution without hardware or sensing dependencies. The repository is publicly available at `https://github.com/luca-bettermann/pred-fab-mock/tree/release/isarc-2026`.

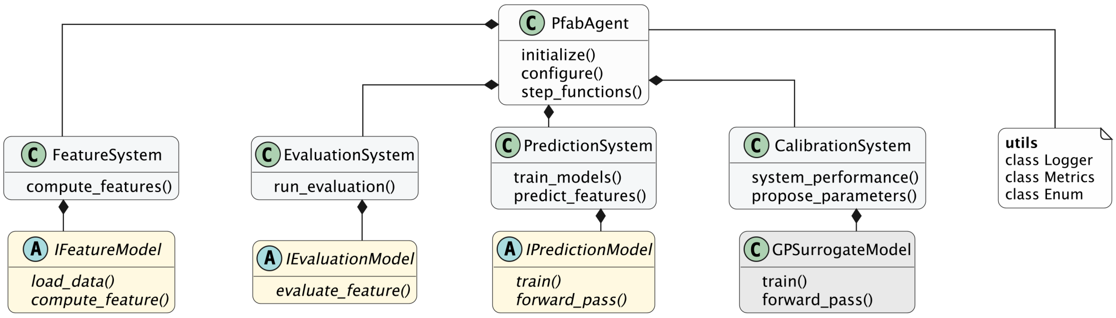

Figure 3. Schematic illustration of the modular `PfabAgent` system architecture.

3.5 Cross-layer integration

The predictive layer described in Sec 3.3 is self-contained and constitutes the methodological scope of this paper. However, as indicated in Fig. 1, the PFAB architecture is explicitly designed to leverage the predictive capabilities in preceding layers of the digital fabrication pipeline. These integrations reuse the foundational data structures and agent abstractions described in this section, and are briefly discussed to clarify the architectural intent of PFAB.

Fabrication-time adaptation. In principle, predictive models trained within the PFAB workflow can be fine-tuned and queried incrementally during fabrication as new *as-fabricated* data becomes available. By operating on explicit dimensions (e.g., printing layers), such integration would enable localized parameter recalibration and predictive closed-loop control. While architecturally supported, the methodology of fabrication-time adaptation is not detailed in this paper.

Design-time prediction. Similarly, trained prediction and evaluation models can be integrated in the design layer to estimate expected features and performance without physical execution. This enables outcome-informed design decisions, but remains outside the scope of this work.

4 Methods

The PFAB architecture is separated into framework-level orchestration and user-defined fabrication logic, as illustrated by Fig. 3. The `PfabAgent` serves as the single entry point and coordinates exploration, training and inference by delegating execution to specialized orchestration systems. These systems automatically manage any number of feature, evaluation and prediction components while keeping the core workflow unchanged.
Fabrication-specific logic is integrated through the highlighted abstract interfaces. Feature extraction is implemented via the `IFeatureModel`, which transforms raw sensor data into structured features. The `IEvaluationModel` maps these features to normalized performance attributes. Predictive models defined through the `IPredictionModel` contract are trained on the resulting dataset and queried during calibration. Together, these interfaces define the low-level extension points of PFAB, enabling fabrication-specific implementations while preserving a fixed orchestration flow.

4.1 Dimensional Domains

In digital fabrication, scalar process parameters are defined per experiment, whereas sensor-derived features are inherently dimensional, measured at specific positions within the fabrication structure (e.g., per layer, per path segment, or per structural node). PFAB formalizes this through the concept of a *Domain*: a named, ordered iteration space composed of *Dimension* objects, each representing one structural axis of the fabrication system (Listing 1). Features declare the domain they belong to, clearly defining their dimensional structure. The schema enforces that all user-defined models contain a coherent domain declaration, validated at system initialization (Listing 2). This explicit structure makes the agent's systems directly tailored to the structural element being fabricated, and forms the architectural basis for fabrication-time adaptation and predictive closed-loop control.

4.2 Calibration strategy

PFAB treats parameter selection as a constrained numerical optimization problem executed by a single solver orchestrated by the `CalibrationSystem`. In both the exploration and inference modes, the system calibrates process parameters by solving the same optimization problem, differing only in the objective function being optimized.
Calibration objectives are expressed through the scalar system performance $S(\cdot)$ that aggregates multiple task-specific performance attributes into a single normalized score. Each attribute is assumed to be normalized to $[0, 1]$

and assigned a user-defined weight w_i reflecting its relative importance. The system performance is computed as

$$S(\mathbf{a}) = \frac{\sum_i w_i\, a_i}{\sum_i w_i}, \tag{1}$$

where $\mathbf{a}$ denotes the vector of performance attributes.

4.2.1 Active learning objective

During dataset generation, calibration is driven by an active learning approach that reduces epistemic uncertainty while maintaining alignment with the user-defined system performance. PFAB employs a lightweight Gaussian Process-based surrogate model `GPSurrogateModel` that maps input parameters directly to expected performance attributes, while providing the required uncertainty estimates. The acquisition function follows an Upper Confidence Bound (UCB) formulation [18]:

$$\alpha(\mathbf{x}) = (1-\kappa)\, S\big(\boldsymbol{\mu}(\mathbf{x})\big) + \kappa\, S\big(\boldsymbol{\sigma}(\mathbf{x})\big), \tag{2}$$

where $\boldsymbol{\mu}(\mathbf{x})$ and $\boldsymbol{\sigma}(\mathbf{x})$ denote the predicted mean and uncertainty, respectively, and $\kappa \in \big[0,1\big]$ controls the exploration–exploitation balance *within the exploration mode*. The aggregation function $S(\cdot)$ is applied consistently to both quantities to prioritize information gain in regions relevant to the task objective.

4.2.2 Performance-driven objective

In the inference mode, calibration switches from the surrogate model to the full prediction pipeline to prioritize accuracy over uncertainty. The `PredictionSystem` maps input parameters $\mathbf{x}$ to predicted physical features $\mathbf{y}$, which are scored by the `EvaluationSystem` and aggregated into a scalar system performance:

$$J(\mathbf{x}) = S\big(E\big(f_\theta(\mathbf{x})\big)\big)\,, \tag{3}$$

where f_θ denotes the prediction model and $E(\cdot)$ the evaluation mapping.

5 Case Study: Learning–by–Printing

While PFAB is designed as a fabrication-agnostic predictive framework, this paper grounds its concepts through an instantiation in extrusion-based additive manufacturing. In this context, predictive fabrication manifests as Learning–by–Printing (LbP), in which fabrication outcomes are evaluated and used to calibrate process parameters such as nozzle speed and layer height. PFAB provides the structured data representation, learning pipeline, and calibration logic required to implement the fabrication–learning loop (Fig. 2).

5.1 Proof of Concept

The feasibility of Learning–by–Printing was demonstrated in a prior study on data-driven parameter calibration for extrusion-based 3D printing in construction [19]. That work aimed to validate whether a physical printing system can autonomously improve fabrication quality with respect to a defined objective using sensor data and predictive models.

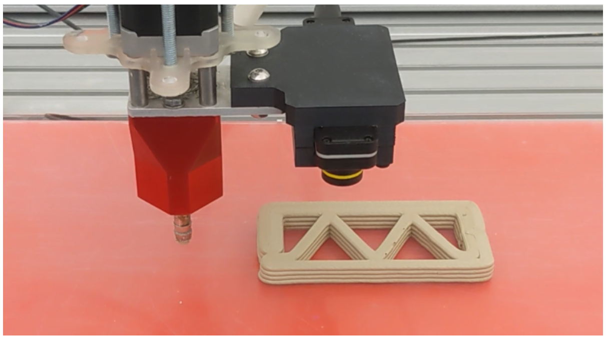

Figure 4. Extrusion-based clay 3D printing setup used in the Learning–by–Printing proof of concept [19].

The study executed the fabrication–learning loop in inference mode. A prediction model was trained from structured fabrication data and subsequently used to calibrate process parameters for future prints. Repeated fabrication trials demonstrated measurable improvements in geometric accuracy and structural consistency, validating the practical effectiveness of embedding predictive models directly into the fabrication workflow.
The Learning–by–Printing study is implemented using PFAB's schema-first formulation. Table 2 summarizes the parameters, features, and performance attributes used in the case study.

Table 2. Schema instantiation in the Learning–by–Printing case study [19].

Parameters	Features	Performance Attrs.
layer time	node overlap	p_{node}
layer height	filament width	$p_{filament}$
path offset	print time	$p_{process}$

Fabrication logic was implemented by extracting geometric features from computer vision-based models and evaluating them against task-specific target values derived from the intended print geometry. These steps directly realize PFAB's `IFeatureModel` and `IEvaluationModel` interfaces, while the `IPredictionModel` was instantiated using Gaussian Process Regression.
The calibration objective is formulated as the weighted

system performance,

$$S(\mathbf{a}) = \frac{w_1 * p_{node} + w_2 * p_{filament} + w_3 * p_{process}}{w_1 + w_2 + w_3}, \quad (4)$$

with weights $w_1 = 3.0$, $w_2 = 2.0$, and $w_3 = 1.25$ reflecting the relative importance of the performance terms. Optimizing this objective enabled automatic parameter calibration toward improved fabrication quality without any manual intervention.

6 Discussion and Conclusion

Prior research in data-driven digital fabrication has established sensing, feature extraction, prediction, and monitoring as key enablers for improving process reliability. However, these contributions typically address isolated aspects of the fabrication pipeline and remain tightly coupled to specific machines, materials, or experimental setups. This fragmentation limits their applicability to first-time-right manufacturing beyond the predefined setting of individual studies. Robust digital fabrication therefore requires jointly addressing process complexity and sensitivity to local, context-dependent disturbances in a generalizable manner.

PFAB contributes a unifying software framework that formalizes predictive fabrication as a structured, end-to-end process linking data acquisition, learning, and parameter calibration. By enforcing consistent data structures and modular interfaces, PFAB enables the development of predictive models that capture general fabrication behavior across experiments. The presented proof of concept in extrusion-based additive manufacturing demonstrates that such models, when embedded directly into the fabrication workflow, can guide parameter calibration and measurably improve fabrication outcomes without manual intervention.

At a conceptual level, PFAB addresses fabrication complexity by modeling the interactions between design, material, process, and environment in a data-driven manner. While this paper focuses on the predictive layer and its calibration workflow, the framework's architecture is explicitly designed to support fabrication-time adaptation as well as design-time prediction. Predictive closed-loop control targets local, non-generalizable sensitivities during execution, while an outcome-informed design stage promotes desirable fabrication outputs. In this way, PFAB provides a foundation for systematically addressing both complexity and sensitivity across the layers of digital fabrication systems.

The framework is presented as a conceptual and architectural contribution referencing an initial proof of concept; extensive empirical validation will be reported in upcoming publications. In practice, predictive performance remains bounded by the quality of feature extraction and evaluation criteria, which cannot be fully standardized and must remain customized to the fabrication context. Extending the predictive layer to fabrication-time adaptation introduces constraints on sensing latency and computational cost, which scale with the complexity of the prediction model's forward pass. These challenges are intrinsic to digital fabrication and reinforce the need for a system-level framework that makes such dependencies explicit and manageable. By standardizing how data, models, and fabrication systems interact, PFAB promotes modularity, transferability, and reproducibility across fabrication contexts. We argue that such data-driven frameworks form a necessary foundation for advancing digital fabrication toward reliable, first-time-right manufacturing in construction.

Acknowledgement

The authors gratefully acknowledge the funding received from EU Pathfinder Project AM2PM (Grant Agreement No. 101162318) as well as from German Research Foundation (DFG) in the frame of Transregio 277/2 project (project number 414265976).

References

[1] Alexander Paolini, Stefan Kollmannsberger, and Ernst Rank. Additive manufacturing in construction: A review on processes, applications, and digital planning methods. *Additive manufacturing*, 30:100894, 2019.

[2] R. Wolfs. The status quo of 3d concrete printing: are we there yet? *RILEM Technical Letters*, 8:182–189, 2023.

[3] Richard A Buswell, WR Leal De Silva, Scott Z Jones, and Justin Dirrenberger. 3d printing using concrete extrusion: A roadmap for research. *Cement and concrete research*, 112:37–49, 2018.

[4] X. Zhang, D. Chu, X. Zhao, C. Gao, L. Lu, Y. He, and W. Bai. Machine learning-driven 3d printing: A review. *Applied Materials Today*, 39:102306, 2024.

[5] M. Slepicka and A. Borrmann. Fabrication information modeling for closed-loop design and quality improvement in additive manufacturing for construction. *Automation in Construction*, 168:105792, 2024.

[6] R. A. Buswell, W. R. Leal da Silva, F. P. Bos, H. R. Schipper, D. Lowke, N. Hack, H. Kloft, V. Mechtcherine, T. Wangler, and N. Roussel. A process classification framework for defining and describing digital fabrication with concrete. *Cement and Concrete Research*, 134:106068, 2020.

[7] R. Wolfs, D. Bos, J.-F. Caron, M. Gerke, R. Mesnil, R. Buswell, N. Ducoulombier, N. Hack, E. Keita, P. Kinnell, et al. On-line and in-line quality assessment across all scale levels of 3d concrete printing. *Cement and Concrete Research*, 185:107646, 2024.

[8] J. Versteege, R. J. M. Wolfs, and T. A. M. Salet. Data-driven additive manufacturing with concrete: Enhancing in-line sensory data with domain knowledge, part i: Geometry. *Automation in Construction*, 172:106020, 2025.

[9] J. Versteege, R. J. M. Wolfs, and T. A. M. Salet. Data-driven additive manufacturing with concrete: Enhancing in-line sensory data with domain knowledge, part ii: Moisture and heat. *Automation in Construction*, 177:106327, 2025.

[10] S. Senthilnathan and B. Raphael. Predicting buildability using the surface texture of 3d printed concrete elements. *Journal of Architectural Engineering*, 31(2):04025010, 2025.

[11] H. Zhao, X. Wang, Z. Chen, X. Liu, Y. Wang, J. Wang, and J. Sun. Microcrack investigations of 3d printing concrete using multiple transformer networks. *Automation in Construction*, 172:106017, 2025.

[12] S.-Y. Geng, B.-Y. Cheng, W.-J. Long, Q.-L. Luo, B.-Q. Dong, and F. Xing. Co-driven physics and machine learning for intelligent control in high-precision 3d concrete printing. *Automation in Construction*, 176:106294, 2025.

[13] H. Zhao, J. Sun, X. Wang, Y. Wang, Y. Su, J. Wang, and L. Wang. Real-time and high-accuracy defect monitoring for 3d concrete printing using transformer networks. *Automation in Construction*, 170: 105925, 2025.

[14] A. Maissen, A. A. Apolinarska, S. V. Kuhn, L. Salamanca, M. A. Kraus, K. Tatsis, G. Casas, R. Bischof, R. Rust, W. Kaufmann, et al. Aixd: Ai-extended design toolbox for data-driven and inverse design. *Computer-Aided Design*, page 103945, 2025.

[15] R. S. Olson and J. H. Moore. *TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning*, pages 151–160. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05318-5. doi:10.1007/978-3-030-05318-5_8. URL `https://doi.org/10.1007/978-3-030-05318-5_8`.

[16] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter. Auto-sklearn 2.0: Hands-free automl via meta-learning. *Journal of Machine Learning Research*, 23(261):1–61, 2022.

[17] L. Bettermann. PFAB: Predictive Fabrication Framework (v0.1.0), December 2025. URL `https://github.com/luca-bettermann/pred-fab`.

[18] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1015–1022, Madison, WI, USA, 2010. Omnipress.

[19] L. Bettermann, M. Slepicka, S. Esser, and A. Borrmann. Data-driven parameter calibration in additive manufacturing for construction. In *Proc. of the 32nd Int. Conference on Intelligent Computing in Engineering (EG-ICE)*, 2025.