A parameter-aware graph neural network for estimation of manufacturing resources in additive manufacturing

The International Journal of Advanced Manufacturing Technology (2026) 144:811–825
https://doi.org/10.1007/s00170-026-17892-2

ORIGINAL ARTICLE

A parameter-aware graph neural network for estimation of manufacturing resources in additive manufacturing

Niccolò Giovenali[1] · Giulia Bruno[1] · Paolo Chiabert[1] · Frédéric Segonds[2]

Received: 26 September 2025 / Accepted: 13 March 2026 / Published online: 30 March 2026

Abstract

Additive Manufacturing (AM) offers unprecedented design freedom, yet evaluating the manufacturing resources required for complex geometries remains a significant computational bottleneck during the early design phase. Traditional slicing software, while accurate, is too slow to support high-frequency iterative workflows such as Generative Design. To address this, we propose the Parameter-Aware Geometric Estimator (PAGE-Net), a Graph Neural Network (GNN) framework designed to instantly predict Life Cycle Inventory (LCI) data–specifically Part Mass, Support Mass, and Total Print Time–directly from raw 3D meshes. Unlike existing voxel-based deep learning methods that suffer from discretization errors or static parameter assumptions, PAGE-Net leverages Feature-Steered Graph Convolutions (FeaStConv) to extract topological features from the native mesh while dynamically incorporating user-defined printing parameters (e.g., infill density, layer height). Trained and validated on a comprehensive dataset of approximately 90,000 geometries using a robust 3-fold cross-validation scheme, the model achieves high predictive accuracy, with $\boldsymbol{R^2}$ scores exceeding 0.96 for material and time estimation. Computational benchmarks demonstrate an average inference time of 77 milliseconds per object–offering a speedup of approximately $\mathbf{30}\times$ compared to optimized command-line slicing. By providing near real-time feedback on manufacturing resources, this framework serves as a critical enabler for data-driven Eco-Design and automated topology optimization.

Keywords Additive manufacturing · Machine learning · Graph neural network · Eco design

1 Introduction

Additive Manufacturing (AM), particularly Fused Filament Fabrication (FFF), has revolutionized production by enabling mass customization and the fabrication of complex geometries without the constraints of traditional tooling. This freedom offers immense potential for Eco-Design, allowing engineers to minimize material usage through topology optimization and lightweighting. However, the maxim that AM offers "complexity for free" is not absolute. As demonstrated by Pradel et al. [13], increased geometric intricacy correlates non-linearly with manufacturing time and material consumption, revealing that complex designs do incur distinct environmental and economic costs.

Consequently, a significant bottleneck in the sustainable adoption of AM is the lack of immediate feedback regarding these costs during the early conceptual phase. Life Cycle Assessment (LCA) is the standard for quantifying environmental impact, but it relies on precise Life Cycle Inventory (LCI) data–specifically mass and process energy (time).

Giulia Bruno, Paolo Chiabert and Frédéric Segonds contributed equally to this work.

✉ Niccolò Giovenali
niccolo.giovenali@polito.it

Giulia Bruno
giulia.bruno@polito.it

Paolo Chiabert
paolo.chiabert@polito.it

Frédéric Segonds
Frederic.SEGONDS@ensam.eu

[1] DIGEP, Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italia

[2] LCPI, Arts et Métiers Institute of Technology, 151, Boulevard de l'Hôpital, 75013 Paris, France

Currently, obtaining this data requires simulating the toolpath using slicing software. While accurate for final production, slicing acts as a computationally expensive "black box" that disrupts the digital design workflow. This latency creates a cognitive gap: designers are often forced to make geometric decisions without knowing how subtle variations or printing parameters will influence the final environmental footprint.

This lack of visibility becomes even more critical in modern automated workflows, such as Generative Design (GD) and Topology Optimization, where the design space expands exponentially. Recent literature highlights the growing effectiveness of hybrid metaheuristic approaches to navigate these complex spaces, coupling Artificial Neural Networks (ANNs) with bio-inspired algorithms. Notable examples include the Supercell Thunderstorm Optimizer [14], the Superb Fairy-Wren algorithm [9], and the Catch Fish Optimization Algorithm [4], applied to optimize industrial components ranging from heat exchangers to electric vehicle battery boxes. Similarly, bio-inspired methods have been utilized for vehicle suspension components [15] and crashworthiness analysis of auxetic structures [25].

However, the efficacy of these advanced generation methods is constrained by the cost of evaluation. While algorithms can propose thousands of lightweight variants, assessing the actual manufacturability and resource consumption for each variant via traditional slicing is practically infeasible. This leaves a blind spot in the optimization loop, where the theoretical optimum may differ from the manufacturable eco-optimum. As noted in a recent comprehensive review of ML in AM [18], current applications are heavily skewed towards in-situ process monitoring, leaving a significant gap in predictive tools specifically designed to empower decision-making in the early product design phase.

1.1 Limitations of existing predictive models

To mitigate these computational costs, data-driven surrogate models have been proposed. It is important to note that while the broader literature on metaheuristics and AM optimization is extensive, a systematic search on Scopus reveals that research specifically targeting the rapid, pre-slicing prediction of LCI data from raw 3D geometries remains sparse. Existing studies comparable to the proposed approach primarily rely on either manually engineered features or voxel-based approaches, both of which present distinct limitations for Design for Additive Manufacturing (DfAM).

Traditional machine learning models, such as Artificial Neural Networks (ANNs), have been trained on curated sets of high-level geometric features to predict FFF build time. For instance, recent studies [16] developed an ANN using a small dataset of 43 CAD models, exploring its application on a vector of Representative Features (e.g., bounding box dimensions, volume, surface area). While effective for simple shapes, such feature-based methods depend heavily on manual pre-selection of descriptive features, which often fail to capture the topological nuances of organic generative designs.

To automate feature extraction, researchers have turned to Deep Learning, specifically 3D Convolutional Neural Networks (CNNs) that operate on voxelized representations. A voxel, or volumetric pixel, represents a value on a regular grid in three-dimensional space. Pioneering research by Williams et al. [24] developed a build metric prediction using a large, synthetically generated repository of 18,000 simple parametric shapes. A direct comparison of ANN and CNN methods by Oh et al. [11] utilized a more realistic dataset of approximately 3,000 models. Their findings conclusively demonstrated that CNNs operating on voxel representations consistently outperformed feature-based models.

Despite their advantages, these voxel-based approaches face three critical limitations that hinder their use:

- Discretization Error: Voxelization inherently introduces a loss of fine geometric features due to its gridded nature. High-resolution grids required to capture thin walls or smooth curves (common in AM) result in prohibitive memory costs.
- Static Parameters: The aforementioned studies typically predict outcomes for a fixed set of printing parameters (e.g., a single layer height). This restricts their use as interactive tools, as designers cannot evaluate how changing process settings affects the outcome.
- Dataset Scope: Existing datasets are often composed of simplistic geometries or represent filtered subsets that do not fully capture the complexity of real-world functional parts.

1.2 Proposed contribution: PAGE-Net

To address these limitations, we propose the Parameter-Aware Geometric Estimator (PAGE-Net), a Graph Neural Network (GNN) framework that operates directly on the native triangular mesh representation (STL) of the 3D object.

By treating the mesh as a graph, the network learns geometric features directly from the high-fidelity surface topology, avoiding the information loss of voxelization. Crucially, unlike previous works that predict for a fixed configuration, PAGE-Net is parameter-aware: it inherently accepts a vector of key printing parameters (e.g., layer height, infill density) as an additional input, transforming the model from a static estimator into a dynamic decision-support tool.

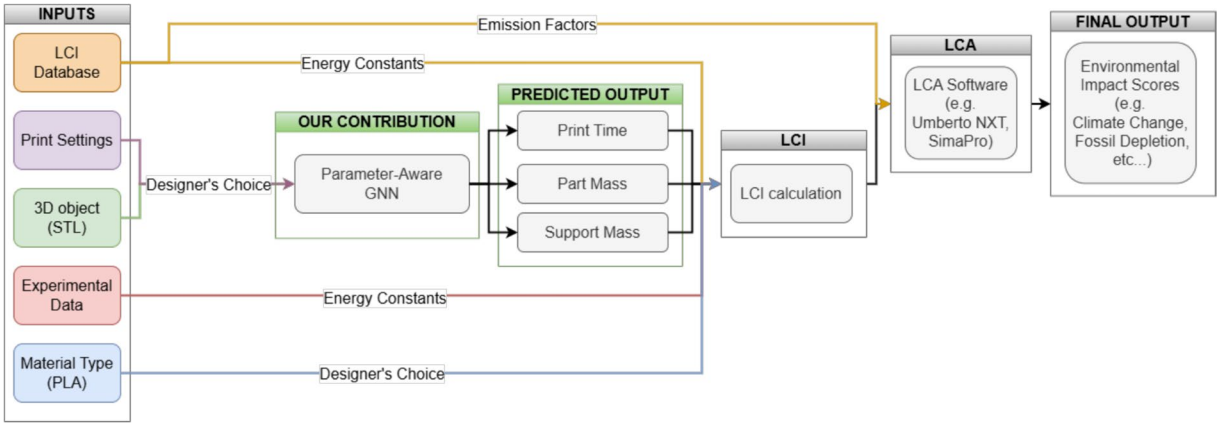

Fig. 1 Conceptual Framework for Data-Driven Eco-Design. This paper focuses on the "Predictive Module" (highlighted), utilizing a GNN to instantly estimate manufacturing resources (Mass, Time). These predictions constitute the Life Cycle Inventory (LCI), enabling the downstream integration of LCA databases for environmental assessment

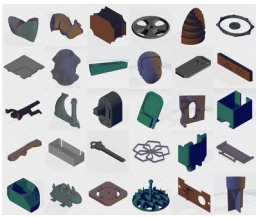

Fig. 2 Representative subset of geometries from the Slice-100K dataset

While this study does not perform a full LCA (i.e., impact characterization), it solves the upstream bottleneck by providing the essential Life Cycle Inventory (LCI) data required for such assessments in real-time. Figure 1 illustrates how this predictive module fits within a broader Eco-Design workflow, effectively translating abstract geometric data into tangible manufacturing resources that serve as the foundation for future environmental impact calculations.

2 Methodology

2.1 Dataset generation

2.1.1 Data source and geometric selection

We utilized the Slice-100K dataset [6], a comprehensive collection of additively manufacturable models derived from Objaverse-XL [2] and the Thingi10k dataset [26]. An example of the objects can be seen in Fig. 2.

To ensure the topological quality required for GNN message passing, we curated a subset of valid geometries through a Quality-First Filtering Strategy:

1. Dimensional Consistency: To eliminate legacy unit scaling artifacts (e.g., inch-to-mm conversion errors), we enforced a minimum volume threshold of $1\,\mathrm{cm}^3$.
2. Watertightness: Using the `trimesh` library, we discarded non-manifold meshes or those containing holes to prevent slicing failures.
3. Simplification and Node Constraint: High-resolution meshes were decimated to a target of ≈4,000 faces using Quadratic Error Metrics (QEM) [3], balancing geometric fidelity with computational feasibility. Crucially, geometries that were too complex to be simplified to this target without topological collapse (or those retaining excessive node counts) were discarded to ensure a consistent graph size for the network.
4. Topological Validation: We finally removed graph artifacts such as isolated nodes or self-loops that disrupt gradient flow.

This pipeline yielded 9,049 unique geometries (Fig. 3).

2.1.2 Definition of the printing parameter space

The printing parameters were selected based on their established influence on energy consumption and final object quality, allowing the tool to explore critical design-manufacturing trade-offs. As highlighted in recent literature, print time is a dominant factor in energy consumption [21], yet its reduction via increased Printing Speed or Layer Height typically compromises dimensional accuracy and interlayer adhesion [10]. Conversely, the structural robustness of the component is primarily governed by the Infill Density and the number of Perimeters (external walls).

Consequently, we grouped these variables into two high-level categories:

- *Quality-related:* Printing Speed, Layer Height.
- *Robustness-related:* Infill Density, Perimeters.

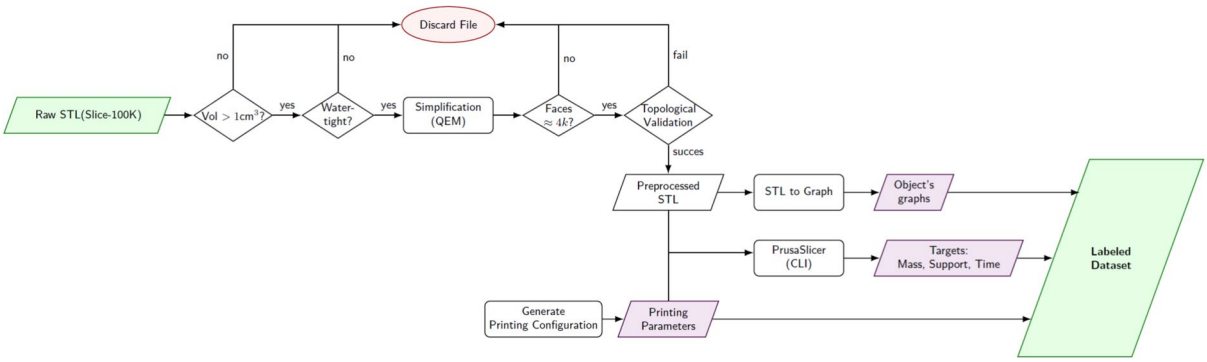

Fig. 3 Dataset preparation pipeline

To define a realistic search space, we analyzed the configuration database within PrusaSlicer, examining standard profiles for various industrial printers (e.g., Prusa MK4S, Ultimaker S7). We established a mapping where a high-level parameter change triggers a coherent modification of the underlying low-level settings (e.g., a “High Speed” setting adjusts approximately 23 distinct acceleration and jerk variables). We then discretized each of the four parameters into three levels (Low, Medium, High) through linear interpolation between identified industry minimums and maximums. This approach results in a full factorial design of $3^4 = 81$ unique printing configurations, covering the spectrum from “Draft” to "High-Fidelity" prints.

2.1.3 Object orientation and support strategy

Two pre-processing decisions were standardized to ensure consistent ground truth generation:

- Orientation: Geometries were processed in their "as-found" orientation. Rather than pre-optimizing alignment, this approach exposes the network to diverse scenarios, training it to generalize across both optimal and sub-optimal orientations.
- Support Generation: We fixed the overhang threshold at 45°. Following a preliminary efficiency study on 1,000 samples, the “Snug” support style was selected as the standard, as it demonstrated superior material efficiency compared to “Grid” or “Organic” styles.

2.1.4 Slicing simulation and data generation

Ground truth labels (Print Time, Material Mass, Support Mass) were generated using the PrusaSlicer CLI (v2.9.2), chosen for its reliability and batch-processing capabilities.

To efficiently cover the parameter space without the prohibitive cost of a full factorial simulation (81 configs $\times \approx 9,000$ shapes), we employed Latin Hypercube Sampling (LHS) [8]. For each geometry, we sampled 10 distinct configurations that uniformly cover the parameter space. The simulation pipeline involved slicing each object twice (with and without supports) to isolate the support mass. The final dataset comprises approximately 90,000 unique input-output instances, providing a dense sampling of the design-manufacturing space.

2.1.5 Input representation and feature engineering

To effectively capture the relationship between object geometry, process parameters, and printing resources, we propose a hybrid input representation that combines local mesh topology with global geometric descriptors.

2.1.6 Local node features

The core input is the raw 3D mesh, represented as a graph $G = (V, E)$. To ensure the network is invariant to absolute spatial positioning while retaining scale information, the mesh is first centered at the origin (0, 0, 0) and scaled such that it fits within a unit sphere $[-1, 1]$.

Each node $v_i \in V$ is then initialized with a 9-dimensional feature vector x_i. In addition to the raw 3D coordinates (x, y, z) and vertex normal vector (n_x, n_y, n_z), we explicitly compute three engineered features to provide the network with manufacturing-specific context:

- *Relative Print Height* ($\tilde{z}$): Calculated as $\tilde{z}_i = (z_i - z_{min})/(z_{max} - z_{min})$ In layer-based manufacturing (FFF), the Z-axis represents the time dimension of the build process. This feature explicitly informs the network of a vertex’s position in the printing sequence, which correlates with structural leverage.
- *Radial Distance* (d_r): Computed as the Euclidean norm $\|\mathrm{p}_i\|_2$ from the object centroid (origin). This captures the spatial distribution of mass, helping the network distinguish between bulk core structures and distant extremities (e.g., overhangs or thin features) that are more prone to vibration and thermal distortion.

- *Discrete Mean Curvature* (κ): We approximate the local surface curvature using the hyperbolic tangent transformation: $\kappa_i' = \tanh(\kappa_i)$. This bounded descriptor allows the GNN to differentiate between flat surfaces–where the print head can reach maximum velocity–and high-curvature regions that require deceleration to maintain precision.

2.1.7 Global geometric features

To augment the local graph representation with macroscopic shape information, we compute a vector of 17 global geometric features. While basic dimensions (volume, area, bounding box extents) provide scale, we explicitly calculate complex shape descriptors to capture topological characteristics that govern manufacturing dynamics. These are defined as follows:

- *Inertial Properties (6 features):* The distribution of mass affects the object's stability during the rapid accelerations of the print head. We compute the inertia tensor I assuming uniform density. The principal moments of inertia (I_1, I_2, I_3) are the eigenvalues of I. To capture the object's orientation relative to the build plate, we compute the Z-alignment features α_i, defined as the absolute dot product between the *i*-th principal axis eigenvector v_i and the vertical build axis $\mathrm{z} = (0, 0, 1)$:

$$\alpha_i = |\mathrm{v}_i \cdot \mathrm{z}| \tag{1}$$

High alignment values indicate that the object's mass is distributed primarily along the vertical axis, potentially reducing wobbling artifacts.

- *Compactness (Sphericity):* This dimensionless coefficient measures how efficiently a shape encloses volume relative to its surface area. It is defined using the isoperimetric quotient:

$$C = \frac{36\pi V^2}{A^3} \tag{2}$$

where V is the mesh volume and A is the surface area. A value of 1.0 indicates a perfect sphere.

- *Volume Efficiency:* This metric quantifies the filling ratio of the object within its axis-aligned bounding box (AABB). It serves as a proxy for the geometric sparsity:

$$E_{vol} = \frac{V_{mesh}}{L \times W \times H} \tag{3}$$

Low volume efficiency typically implies significant empty space within the bounding volume, leading to increased air travel time for the print head as it moves between disconnected or distant islands of geometry.

- *Convexity:* Defined as the ratio of the mesh volume to the volume of its convex hull (V_{hull}):

$$\Omega = \frac{V_{mesh}}{V_{hull}} \tag{4}$$

A low convexity score indicates the presence of significant concavities, deep recesses, or hollow sections. These features strongly correlate with the need for complex travel moves and potentially increased support material generation to handle internal overhangs.

- *Overhang Ratio:* To explicitly inform the network about support requirements, we calculate the proportion of the surface area that exceeds the critical overhang angle (typically $45°$). Let n_f be the normal vector of face *f* and A_f its area. The overhang ratio R_{OH} is given by:

$$R_{OH} = \frac{1}{A_{total}} \sum_{f \in \mathcal{F}} A_f \cdot \mathbb{I}(\mathrm{n}_f \cdot \mathrm{z} < \cos(135°)) \tag{5}$$

where $\mathbb{I}$ is the indicator function. This feature provides a direct heuristic for the volume of support structures required.

- *Mesh Density:* Calculated as the number of faces per unit of surface area (N_f/A). High mesh density often indicates high-frequency geometric details or curvature that requires the printer firmware to process dense G-code streams, potentially inducing velocity bottlenecks.
- *Print Configuration:* Finally, the printing configuration is represented as a 4-dimensional vector comprising the normalized values for Printing Speed, Layer Height, Infill Density, and Perimeter Count (see Section 2.1.2 for details).

2.1.8 Feature scaling and distribution analysis

Raw manufacturing data frequently exhibits highly skewed distributions with long right tails. As illustrated in the left column of Fig. 4, geometric features such as object volume and surface area, as well as the target variable support mass, contain a dense cluster of small values alongside distinct extreme outliers.

Training neural networks on such skewed data is problematic: extreme values can cause gradient instability,

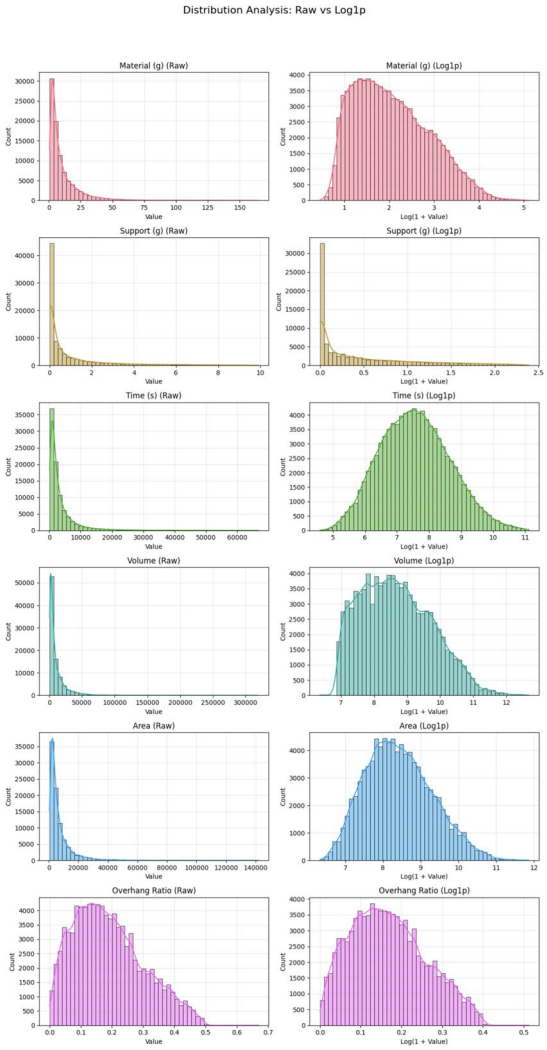

Fig. 4 Some feature distribution analysis. The raw data (left) exhibits severe positive skewness. The application of the $\log(1+x)$ transformation followed by Z-score standardization (right) effectively normalizes the distributions, consistent with Tukey's ladder of powers for variance stabilization

while the compressed bulk of the data leads to poor discrimination in the lower ranges. Following the principles of exploratory data analysis and Tukey's ladder of powers [19], we address this by applying a log-linear transformation, $x' = \log(1+x)$, to all positively skewed features. This transformation was applied to: Volume, Surface Area, Bounding Box Extents, Inertia Tensor components, and the target variables (Material Mass, Support Mass, Print Time). The constant 1 ensures numerical stability for zero-valued entries, such as objects requiring no support material.

Subsequently, to ensure uniform contribution to the gradient descent process, all input and output features–including those previously log-transformed–were standardized using Z-score normalization:

$$z = \frac{x - \mu}{\sigma} \tag{6}$$

where μ is the mean and σ is the standard deviation of the feature across the training set. Features that were naturally bounded by construction (e.g., Compactness, Volume Efficiency, and the normalized printing configuration parameters) were left in their original scale or standardized depending on their variance, ensuring that all network inputs share a comparable dynamic range. The right column of Fig. 4 demonstrates how this pipeline stabilizes the variance, resulting in distributions that are significantly more tractable for optimization.

2.2 Network architecture: PAGE-Net

We introduce PAGE-Net (Parameter-Aware Geometric Estimator), a specialized architecture designed to process unstructured mesh data by leveraging Feature-Steered Graph Convolutions (*FeaStConv*) [20].

Standard Convolutional Neural Networks (CNNs) rely on the regular grid structure of images to associate filter weights with neighbors based on fixed relative positions (e.g., "top-left", "center"). On irregular triangular meshes, such fixed ordering is undefined. While Graph Attention Networks (GAT) address this via attention coefficients, *FeaStConv* generalizes the convolution operator by dynamically determining the correspondence between filter weights and graph neighbourhoods based on the input features themselves. Specifically, instead of assigning a single fixed weight to a neighbour, the layer learns a soft-assignment function $q_m(\mathrm{x}_i, \mathrm{x}_j)$ that maps neighbour j of node i across a set of M weight matrices W_m. The convolution operation is defined as:

$$\mathrm{y}_i = \mathrm{b} + \sum_{m=1}^{M} \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} q_m(\mathrm{x}_i, \mathrm{x}_j) W_m \mathrm{x}_j \tag{7}$$

Crucially, the assignment weights q_m are not static; they are computed dynamically for every edge using a softmax over a linear transformation of the node features[cite: 539]:

$$q_m(\mathrm{x}_i, \mathrm{x}_j) \propto \exp(\mathrm{u}_m^\top \mathrm{x}_i + \mathrm{v}_m^\top \mathrm{x}_j + c_m) \tag{8}$$

where $\mathrm{u}_m, \mathrm{v}_m, c_m$ are learnable parameters. By setting $\mathrm{u}_m = -\mathrm{v}_m$, the network enforces translation invariance in the feature space. This mechanism allows PAGE-Net to effectively detect local geometric patterns (such as curvature changes or sharp edges) regardless of the mesh's tessellation density, mimicking the translation-invariant feature extraction of CNNs on irregular grids.

The architecture (Fig. 5) employs a dual-stream encoder structure to process the heterogeneous input data:

1. *Geometric Stream (Local Encoder):* The 9-dimensional node features are projected into a latent space ($D_0 = 64$) and processed by a stack of 3 *FeaStConv* layers. To facilitate gradient flow and increase model capacity, we employ an expanding channel strategy: the hidden dimension doubles for the first two layers ($64 \rightarrow 128 \rightarrow 256$) and is maintained at the maximum dimension (256) for the final layer. Each layer utilizes $H = 4$ attention heads and includes self-loops to preserve central node information.
2. *Global Aggregation:* Following the convolution block, node features are aggregated into a single graph-level embedding using Attentional Pooling [7]. This mechanism computes a learnable soft-attention score for each node, allowing the model to prioritize geometrically significant regions (e.g., complex overhangs) over flat surfaces when forming the global representation.
3. *Meta-Data Fusion:* The global geometric features (17-dim) and printing configuration (4-dim) are concatenated and processed by a dedicated MLP encoder (Linear $\rightarrow$ BatchNorm $\rightarrow$ LeakyReLU $\rightarrow$ Dropout) to match the dimension of the graph embedding ($D = 256$).

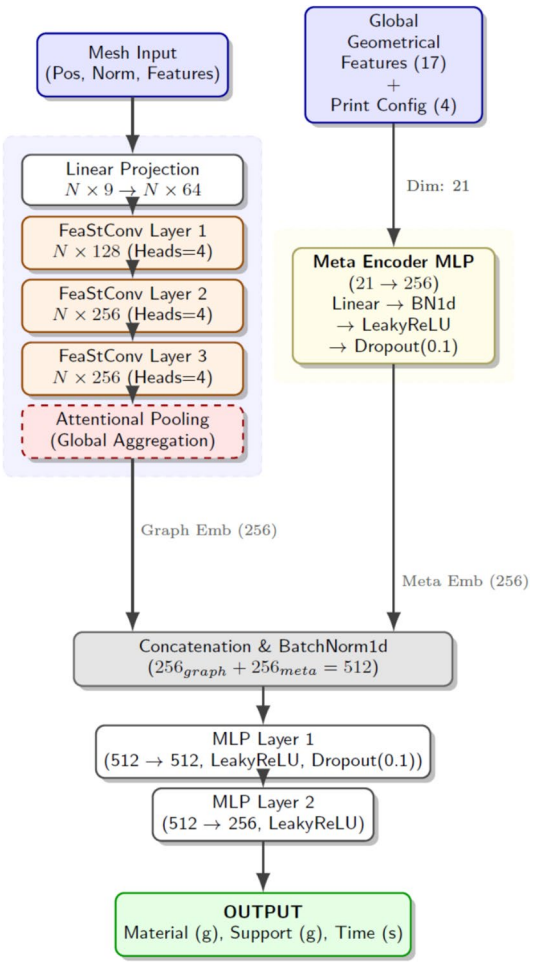

Fig. 5 Architecture of the PAGE-Net model. The framework adopts a two-stream approach: a geometric branch processes the mesh topology via Feature-Steered Graph Convolutions (FeaStConv) and Attentional Pooling, while a parametric branch encodes global printing configurations. The streams are fused via concatenation before the final regression head

The final Fusion Block concatenates the graph embedding and the meta-data embedding. This combined vector is normalized via Batch Normalization to stabilize the feature distribution before entering the decoder. Finally, the representation passes through a regularized decoder MLP (Linear $\rightarrow$ LeakyReLU $\rightarrow$ Dropout $\rightarrow$ Linear $\rightarrow$ LeakyReLU $\rightarrow$ Linear) with hidden sizes $512 \rightarrow 256$ to predict the three target variables.

2.3 Evaluation metrics

To thoroughly assess the predictive capabilities of the trained GNN model, we employed four standard regression metrics that offer complementary insights into model performance: the coefficient of determination (R^2), Mean Absolute Error (MAE), Root Average Squared Error (RASE), and Weighted Absolute Percentage Error (WAPE). The Coefficient of Determination (R^2) quantifies the proportion of variance in the dependent variable that is explained by the model. It provides a primary measure of goodness-of-fit, where a value of 1 indicates perfect replication of the observed outcomes. The Mean Absolute Error (MAE) offers a direct measure of the average magnitude of prediction errors in the physical units of the target variable (grams or seconds). It is calculated as:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{9}$$

where n is the number of samples, y_i is the ground truth, and $\hat{y}_i$ is the predicted value. The Root Average Squared Error (RASE) provides a normalized error metric, expressing the typical prediction error as a proportion of the global range of the dataset. This allows for a comparison of error magnitude across outputs with vastly different scales (e.g., mass vs. time). It is computed as:

$$\mathrm{RASE}(\%) = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{y_{\max} - y_{\min}} \cdot 100 \tag{10}$$

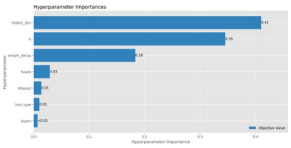

Fig. 6 Hyperparameter Importance (fANOVA). The architectural width (Hidden Dimension) dominates the sensitivity, followed by the Learning Rate and Weight Decay

Finally, to provide a relative error metric that is robust to variations in data scale, we adopted the Weighted Absolute Percentage Error (WAPE). WAPE calculates the sum of absolute errors divided by the sum of actual values:

$$\text{WAPE}(\%) = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i|} \cdot 100 \tag{11}$$

We selected WAPE specifically to address the limitations of the standard Mean Absolute Percentage Error (MAPE) in our domain. While MAPE is commonly cited, it is mathematically unsuitable for datasets containing zero or near-zero values, such as our 'Support' mass (where many geometries require 0g of support). In such cases, the standard MAPE formula ($|y - \hat{y}|/|y|$) yields infinite or exorbitantly high percentage errors for small deviations, disproportionately distorting the aggregate metric. Conversely, WAPE is volume-weighted: it correctly penalizes errors on larger, more significant prints while remaining numerically stable for instances with zero support or small print times.

2.4 Sensitivity analysis

To ensure computational feasibility without compromising statistical representativeness, the optimization was not conducted on the entire sample corpus. Instead, we generated a reduced dataset comprising 30% of the total instances (≈28,500 samples) using stratified sampling [1] based on the discretized target distributions. This ensured that the hyperparameter search space was explored on a population that preserved the critical long-tail and zero-inflated characteristics of the full dataset.

To formalize the selection process, we employed the Tree-structured Parzen Estimator (TPE) algorithm ([22] via the `Optuna` library. This automated optimization study successfully identified a high-performing configuration, achieving a peak mean validation R^2 score of 0.905. To interpret these results, we applied the functional analysis of variance (fANOVA [5]) method provided by `Optuna`'s importance evaluator.

As illustrated in Fig. 6, the analysis revealed a clear hierarchy in hyperparameter sensitivity. The model performance was most sensitive to the Hidden Dimension, followed by the Learning Rate (lr) and Weight Decay. The dominance of the hidden dimension confirms that model capacity is the primary constraint for capturing the complex topology-to-resource mapping.

Once sufficient capacity was established (256 units), the optimizer converged to a relatively aggressive learning rate (3.6×10^{-3}) paired with a moderate weight decay (7.5×10^{-5}), suggesting that the loss landscape is navigable without requiring extreme regularization.

Interesting behaviour was observed regarding network depth: while deeper networks are typically preferred for complex manifolds, the optimization converged to 3 layers rather than the maximum of 4. This indicates that a 3-hop message passing receptive field is sufficient to capture the relevant local topology features for this specific regression task, and adding further layers yields diminishing returns or introduces over-smoothing.

Regarding the loss function, MSE was selected over Huber loss. This suggests that the dataset preprocessing was effective, allowing the model to minimize the squared error directly without needing the robustness of the Huber transition for extreme outliers.

The optimal configuration (Table 1) was subsequently adopted for the full 3-fold cross-validation training.

3 Results and discussion

3.1 Training protocol

The final model was trained using a robust 3-fold cross-validation scheme on the dataset of ≈90,000 samples, ensuring that every geometry is evaluated as unseen data.

Each fold was trained for 30 epochs using the AdamW optimizer with a batch size of 16. Based on the hyperparameter

Table 1 Hyperparameter Search Space and Optimal Configuration identified via TPE optimization (Optuna)

Hyperparameter	*Search Space*	*Distribution*	*Optimal Value*
Architecture			
Hidden Dimension	{64, 128, 256}	Categorical	256
GNN Layers	[2, 4]	Integer	3
Attention Heads	{4, 8}	Categorical	4
Dropout Rate	[0.1, 0.5]	Uniform (Step 0.1)	0.1
Optimization			
Learning Rate	$[10^{-4}, 5 \times 10^{-3}]$	Log-Uniform	3.6×10^{-3}
Weight Decay	$[10^{-5}, 10^{-2}]$	Log-Uniform	7.5×10^{-5}
Loss Function	{MSE, Huber}	Categorical	MSE

sensitivity analysis, we utilized a learning rate of 3.6×10^{-3}, a weight decay of 7.5×10^{-5}, and a dropout rate of 0.1.

To maximize convergence stability, we employed a Cosine Annealing Learning Rate Scheduler. This schedule starts with the initial high learning rate to traverse the loss landscape rapidly and decays efficiently towards zero by the 40th epoch. Unlike step-based decay, this smooth annealing allows the model to settle into wider, more robust minima. We adopted a *save best* checkpointing strategy, retaining the model weights that achieved the highest R^2 score on the validation fold during the training cycle.

3.2 Model predictive performance

The aggregate performance of the PAGE-Net model across the 3-fold cross-validation is summarized in Table 2 and visualized in Fig. 7. The model achieved a strong overall predictive capability with an aggregated mean R^2 of *0.87* and a Weighted Absolute Percentage Error (WAPE) of *12.82%*.

As visualized in the regression scatter plots (Fig. 8), the model exhibits exceptional accuracy in estimating the primary cost drivers: Material Mass and Print Time. With R^2 scores exceeding *0.96* for both targets, the network effectively captures the complex non-linear relationships between the mesh topology, slicing parameters, and G-code generation logic. The extremely low RASE values ($< 1.6\%$) indicate that the error is negligible relative to the global range of the dataset.

3.2.1 Analysis of support estimation

The prediction of Support Mass presents a statistical dichotomy. While relative metrics such as R^2 (≈ 0.68) and WAPE ($\approx 45\%$) suggest a performance gap compared to other

Table 2 Predictive performance metrics (Mean ± Standard Deviation) across 3-fold Cross-Validation. Note the extremely low MAE values relative to the physical scale of the predictions.

Target	*MAE*	R^2 *Score*	*RASE (%)*	*WAPE (%)*
Material (g)	1.11 ± 0.04	0.964 ± 0.005	1.40 ± 0.09	10.59 ± 0.22
Print Time (s)	458.87 ± 17.59	0.961 ± 0.001	1.57 ± 0.03	12.82 ± 0.48
Support (g)	0.43 ± 0.01	0.679 ± 0.038	9.32 ± 0.51	45.36 ± 0.61
Aggregated	-	0.868 ± 0.014	-	12.82 ± 0.48

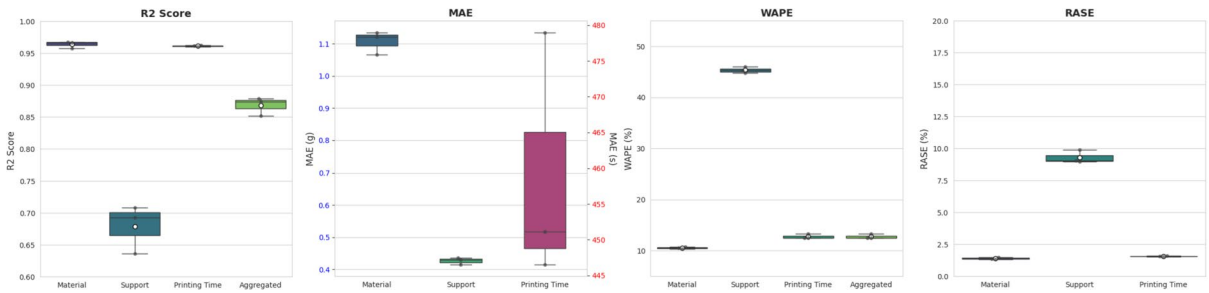

Fig. 7 Box plots showing the distribution of absolute errors for Material, Time, and Support across the 3-fold Cross-Validation on the full dataset. The compact interquartile ranges indicate consistent performance, with outliers (diamonds) representing edge cases in the long-tail distribution

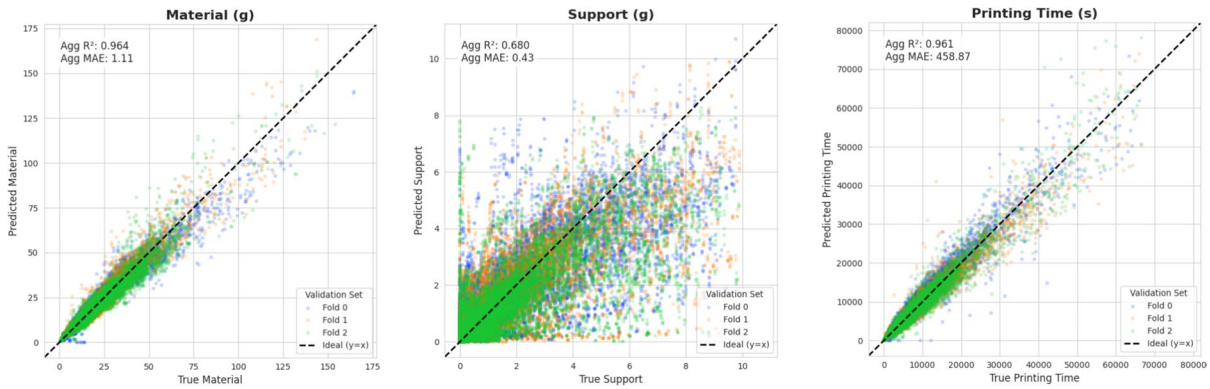

Fig. 8 Regression Scatter Plots for Material, Support, and Time. The dashed line represents the ideal 1:1 prediction, the 3 colours the 3 folds

targets, an analysis of the absolute error reveals a different reality.

The Mean Absolute Error for support is extremely low at 0.43 g. As illustrated in the Support scatter plot (Fig. 8), the data distribution is heavily concentrated near the origin, confirming that the magnitudes involved are physically small. Consequently, an error of 0.43 g–which corresponds to a negligible amount of filament, often less than a single interface layer–can result in high relative penalty despite being inconsequential for practical manufacturing planning.

The discrepancy between the moderate R^2 and the excellent MAE is thus largely attributable to the Zero-Inflated nature of the distribution. A substantial portion of the dataset consists of self-supporting geometries (0g support). In this context, slight fluctuations in the predicted value (e.g., predicting 0.5g instead of 0g) heavily penalize variance-based metrics like R^2.

Therefore, the network successfully solves the implicit dual task:

1. *Detection:* It correctly identifies self-supporting structures in the vast majority of cases (clustering near zero in Fig. 8).
2. *Quantification:* When support is required, the magnitude of the error remains minimal, proving that the model extracts meaningful topological features (overhangs, angles) even if the sparse training signal makes precise regression challenging.

Future iterations could explicitly decouple these tasks via a dual-branch architecture (classification plus regression), but for the purpose of resource estimation, the current precision is already sufficient for effective decision-making.

3.3 Comparative performance discussion

To contextualize the performance of PAGE-Net, Table 3 presents a qualitative comparison with prominent benchmarks in the literature. While direct quantitative comparison is challenging due to disparate datasets, distinct advantages of our approach emerge.

Unlike voxel-based CNN approaches (e.g., [11, 24]), which are typically constrained to fixed printing parameters due to the difficulty of injecting scalar metadata into convolutional grids, PAGE-Net successfully integrates variable process parameters (layer height, infill, etc.). Despite this added complexity, our model demonstrates superior predictive fidelity in terms of goodness-of-fit, achieving R^2 scores (> 0.96 for mass and time) that surpass the voxel-based baselines ($R^2 \approx 0.89$) reported in comparable real-world studies [11]. This validates the hypothesis that operating on the native mesh topology preserves critical geometric details–such as thin walls and exact curvatures–that are lost during voxelization but are crucial for accurate slicer time estimation.

Table 3 Performance Comparison with State-of-the-Art Models

Study	Methodology	Dataset Details	Predicted Target(s)	R^2
PAGE-Net (Our Work)	GNN (Parameter-Aware) on Mesh Data	≈9,000 shapes from Slice-100k. 10 configs x shape. (≈90,000 instances)	Part Mass	*0.96*
			Support Mass	*0.68*
			Build Time	*0.96*
Oh et al. (2021) [11]	3D CNN on Voxel Data	≈3,000 shapes from Thingiverse	Build Time	0.896*
Srikanth et al. (2024) [16]	ANN on Engineered Features	43 specialized CAD models	Build Time	0.880
Williams et al. (2019) [24]	3D CNN on Voxel Data	18,000 synthetic parametric shapes	Part Mass	≈0.99 *
			Support Mass	≈ 0.78*
			Build Time	≈ 0.99*

* Authors applied 4 different transformations (e.g., rotation and/or translation) to their dataset and reported results for various combinations of training and testing transformations. We have reported the results corresponding to their best-case scenario

Furthermore, compared to feature-based ANN methods like [16], which rely on manually extracted descriptors (e.g., bounding box volume), PAGE-Net demonstrates superior scalability. While standard ANNs perform well on small, simple datasets ($R^2 \approx 0.94$), our GNN maintains higher accuracy ($R^2 > 0.96$) on a dataset three orders of magnitude larger, containing highly complex generative shapes. This confirms that the FeaStConv layers successfully extract latent topological features that correlate with filled volume better than simple analytical bounding boxes, specifically for sparse or hollow organic designs.

3.4 Experimental validation

To provide tangible evidence of the proposed framework's applicability and to ground the predictive capabilities of the Graph Neural Network (GNN) model in physical reality, a preliminary experimental validation was conducted. This validation aimed to demonstrate the practical utility of the model by comparing its predictions with actual manufacturing outcomes. We explicitly frame these tests as exemplary case studies, intended to demonstrate real-world applicability alongside the primary computational validation.

Four distinct geometries, corresponding to Parts A, B, C, and D in Table 4, were selected from the dataset. Despite the size constraints of a laboratory setting, these parts were chosen to cover distinct topological challenges:

- Geometries requiring extensive support material (Part C, D) versus self-supporting shapes (Part A, B).
- Compact, dense volumes versus hollow or shell-like structures.
- Variations in infill density and pattern to test the parameter-awareness of the network.

The physical validation was performed within strict time and resource constraints. To ensure scientific rigor, we prioritized statistical reliability over sample quantity or massive scale. Specifically, we performed three repetitions for each physical print to obtain a reliable average print time and exclude outliers. Consequently, selecting significantly larger industrial geometries would have multiplied the machine time beyond current capacity. To compensate for the dimensional limitations of this physical set, we significantly strengthened the computational validation (Section 3.2) by employing a 3-Fold Cross-Validation on the entire dataset of 90,490 geometries. This ensures that the model has been rigorously tested on an extremely diverse range of scales, shapes, and sizes well beyond these four physical prototypes.

Table 4 Experimental Test Results on 3D Printed Parts

Category	*Metric*	*Part A*	*Part B*	*Part C*	*Part D*
Time (s)	Mean Measured	19m 44s	19m 49s	18m 45s	82m 35s
	GNN Pred.	18m 8s	21m 30s	21m 4s	69m 20s
	GNN vs Meas. Error (%)	8.11	8.49	15.02	16.04
	GNN vs Meas. Abs. Error	1m 36s	1m41s	2m 29s	13m 15s
Mass (g)	Mean Measured	4.21	14.94	5.39	23.35
	GNN Pred.	5.17	18.24	5.94	20.58
	GNN vs Meas. Error (%)	23	22	10	12
	GNN vs Meas. Abs. Error	0.96	3.30	0.55	2.77
Support (g)	Mean Measured	N/A	N/A	3.40	12.96
	GNN Pred.	N/A	N/A	3.07	8.85
	GNN vs Meas. Error (%)	N/A	N/A	10	32
	GNN vs Meas. Abs. Error	N/A	N/A	0.33	4.11
Support (Qualitative)		x	x	✓	✓
Shape Complexity Coefficient		1.20	0.17	2.58	0.42
Volume (cm^3)		1.2	82.7	4.8	20.1
Printing Configuration*		M,L,M,M	H,L,M,L	H,M,H,L	H,L,M,M

* *L, M, H correspond to Low, Medium, and High settings for the four printing parameters: Speed, Perimeters, Infill, and Resolution, respectively*

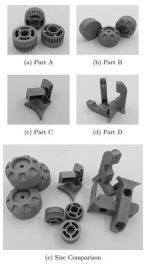

Fig. 9 Photograph of the printed parts for experimental validation. (**a-d**) Individual close-ups of the test geometries. (**e**) Comparative view showing relative scale

Each selected part was additively manufactured using a Prusa MK4S Fused Filament Fabrication (FFF) machine. The actual print time for each fabrication was meticulously recorded using a stopwatch, while the mass of the printed part and, where applicable, the support material, was precisely measured using a laboratory scale with a resolution of 0.01 g. Experimental observations highlighted several factors influencing the total manufacturing time beyond the direct material deposition, such as pre-print routines including initial build plate and nozzle heating, axis calibration, filament purging, and bed leveling. All material mass estimations assume a standard PLA density of 1.24 g/cm^3. A visual representation of the physically printed parts and their relative scale is provided in Fig. 9.

Table 4 provides a summary of the experimental results, including the measured mean for print time and material mass, alongside the calculated percentage errors and absolute errors. The predicted values are averaged over the 3-fold models. The visual comparison of the predicted versus measured values is further illustrated in Fig. 10.

The GNN model's predictions for the experimental parts show varying levels of accuracy. While some percentage errors, particularly for print time, appear relatively high (e.g., 13.52% to 19.65% for Time), it is crucial to consider these in the context of the absolute errors and the intended application of our framework. For instance, an absolute error of 2 to 16 minutes in print time for parts that can take hours (Part D takes 82m 35s) may be acceptable for early design stage Life Cycle Assessment (LCA). Similarly, mass predictions exhibit absolute errors generally well below 1

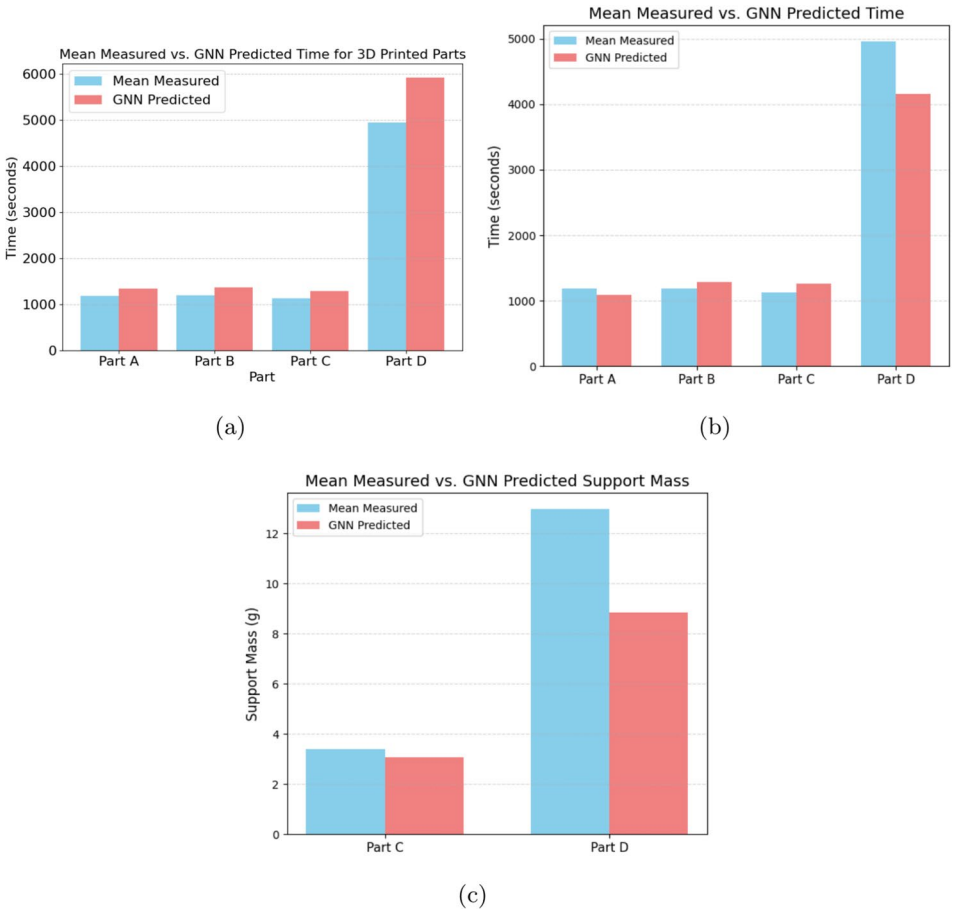

Fig. 10 Visual comparison of Mean Measured versus GNN Predicted values for (**a**) Print Time, (**b**) Part Mass, and (**c**) Support Mass from the experimental validation. Data is derived from Table 4

gram for most parts, except for Part C's mass (3.55g). For the broader purpose of assessing environmental impact during conceptual design, where relative changes and overall trends are often more critical than absolute pinpoint accuracy, the model still offers substantial value for preliminary eco-design exploration.

These experimental results highlight the inherent complexities of real-world printing that a predictive model must capture. While our GNN's broad training approach, utilizing diverse STL orientations and a standardized support strategy (Section 2.3), contributes to its strong generalization capabilities across the large dataset ($R^2 > 0.92$, Section 3.2), a small experimental validation set may reveal higher percentage errors on specific instances due to physical manufacturing variability or challenging edge cases. Nevertheless, the model's consistent performance on the extensive training data, coupled with these practical demonstrations, confirms its utility as a dynamic decision-support tool in eco-design workflows. We hope this publication will stimulate further collaborative research to validate the framework on industrial-scale components in the future, while current work focuses on refining accuracy through network fine-tuning and rigorous outlier detection strategies.

3.5 Computational efficiency and time analysis

To quantify the efficiency gains of the proposed framework, we benchmarked the PAGE-Net inference time against the ground-truth slicing simulation time. To ensure a fair comparison simulating a standard engineering environment, all tests were performed on a laptop equipped with an 11th Gen Intel Core i7-1165G7 processor (4 cores, 2.80 GHz), 16 GB of RAM, and Intel Iris Xe integrated graphics.

Slicing times were measured using PrusaSlicer in Command Line Interface (CLI) mode to exclude the overhead of the graphical user interface (GUI) rendering, representing the theoretical lower bound of the traditional method. Conversely, the GNN inference time includes the data pre-processing (graph construction) and the forward pass. The results on the test dataset ($N = 90, 490$) are detailed in Table 5.

Table 5 Computational time comparison between PrusaSlicer (CLI mode) and PAGE-Net Inference on a consumer-grade laptop. The GNN offers a speedup factor of approximately 30×

Method	*Mean (s)*	*Median (s)*	*Std. Dev (s)*
PrusaSlicer	2.33	2.12	1.86
PAGE-Net	0.077	0.067	0.062

The benchmark reveals that the GNN provides a mean inference time of just 77 milliseconds, offering a speedup of roughly 30x compared to the optimized CLI slicing process. While a 2.3-second slicing time might appear efficient for individual evaluations, this comparison highlights two critical implications for Design for Additive Manufacturing (DfAM) workflows:

- Generative Design Scalability: In Generative Design loops requiring thousands of evaluations, the cumulative time for slicing becomes prohibitive, whereas the GNN completes the task in seconds, acting as a viable surrogate model.
- Reducing Operational Latency: We clarified that the 2.33s benchmark represents an idealized, automated command-line execution. In a typical manual workflow, the effective time-to-feedback is significantly longer due to the context switching required (exporting files, launching slicing software, and manual configuration). By offering inference speeds in the order of milliseconds, our framework conceptually enables a near real-time feedback loop. This opens the possibility for future integrations directly within CAD environments, similar to a spell-checker, where users could potentially receive manufacturability estimates without interrupting their design process.

3.6 Future developments

While the proposed Graph Neural Network framework provides a working basis for the estimation of manufacturing resources in FFF processes, several avenues for future research remain to extend its applicability.

A natural progression of this research involves expanding the model's capabilities within the FFF domain before exploring broader applications. Future iterations could evolve from evaluating discrete parameter categories to incorporating a fully continuous spectrum for all process settings. Additionally, the framework's robustness could be further enhanced by introducing the specific FFF machine profile as an input variable, utilizing training datasets generated by various commercial slicers from different hardware brands.

From an algorithmic perspective, a promising enhancement identified during this study concerns the network's architecture for support structure estimation. The model could benefit from a dedicated classification branch designed to first evaluate whether a geometry strictly requires support, before regressing the corresponding mass. This two-fold approach would likely improve the prediction accuracy for self-supporting features.

Finally, the underlying concept of this framework holds potential for application to other additive manufacturing technologies, such as Stereolithography (SLA) or Selective Laser Sintering (SLS). Since these methods share the fundamental layer-by-layer fabrication principle—which largely dictates the relationship between process parameters, build time, and material usage—the proposed GNN approach remains highly relevant. Nevertheless, such an extension would require careful preliminary studies to properly encode the unique physics, additional variables, and specific constraints inherent to these different manufacturing processes.

Early design decisions are widely acknowledged to determine a substantial portion of a product's total life cycle impact [17]. To fully realize the potential of the proposed framework, future work will focus on integrating the GNN predictions into a formal Life Cycle Assessment (LCA) pipeline. In LCA terminology, the outputs of our model (part mass, support mass, and build time) represent the dynamic, part-specific foreground data of the Life Cycle Inventory (LCI). To translate these variables into actual environmental impacts, they must be coupled with static background data [23]. This involves multiplying the predicted mass by the material's embodied energy, and the predicted time by machine-specific power consumption profiles [12].

A long-term vision involves deploying this integrated GNN-LCA model as a real-time "eco-advisor" plug-in within commercial CAD environments. By bypassing the traditional CAM slicing bottleneck, this setup would provide designers a sustainability feedback as they alter geometry or printing parameters. Furthermore, this rapid evaluation engine could serve as a dynamic fitness function for Generative Design algorithms, allowing them to actively minimize the carbon footprint of complex assemblies. Ultimately, future multidisciplinary studies could investigate how such accessible feedback practically influences human decision-making, supporting a broader transition toward conscious Eco-Design practices.

4 Conclusion

This study introduced a parameter-aware Graph Neural Network (GNN) framework tailored for the estimation of manufacturing resources in Additive Manufacturing (AM). By operating directly on the triangular mesh (STL) representation of 3D objects, the proposed architecture avoids the computational overhead of B-Rep processing and the geometric discretization issues associated with voxelization. A key aspect of this work is the late fusion strategy, which integrates variable printing parameters into the graph embedding, allowing the model to act as a process-aware surrogate model.

Validation on a dataset derived from mechanical and structural components demonstrated the model's predictive capability for part mass, support mass, and build time. The GNN inference operates in the order of milliseconds, offering a noticeable speedup compared to standard slicing simulations. This computational efficiency addresses one of the current bottlenecks in Design for Additive Manufacturing (DfAM), making the framework a highly suitable candidate for integration into automated, high-volume optimization loops such as Generative Design, and paving the way for resource-aware decision-making in engineering design.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

Declarations

Financial Interests The authors have no relevant financial or non-financial interests to disclose.

Consent for Publication This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 352/2022.

References

1. Cochran WG (1977) Sampling Techniques, 3rd edn. John Wiley & Sons
2. Deitke M, Liu R, Wallingford M et al (2023) Objaverse-XL: A Universe of 10M+ 3D Objects. In: 37th Conference on neural information processing systems (NeurIPS 2023) Track on Datasets and Benchmarks
3. Garland M, Heckbert PS (1997) Surface Simplification Using Quadric Error Metrics. In: SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp 209–216. https://doi.org/10.1145/258734.258849. http://www.cs.cmu.edu/
4. Gürses D, Mehta P, Sait SM et al (2025) Battery box design of electric vehicles using artificial neural network-assisted catch fish optimization algorithm. Materialpruefung/Materials Test 67(9):1463–1475. https://doi.org/10.1515/mt-2025-0075
5. Hutter F, Hoos H, Leyton-Brown K (2014) An Efficient Approach for Assessing Hyperparameter Importance. Tech. rep
6. Jignasu A, Marshall KO, Mishra AK et al (2024) Slice-100K: A Multimodal Dataset for Extrusion-based 3D Printing. https://doi.org/10.48550/arXiv.2407.04180
7. Li Y, Gu C, Dullien T et al (2019) Graph Matching Networks for Learning the Similarity of Graph Structured Objects. In: Proceedings of the 36th international conference on machine learning, Long Beach, California
8. Mckay MD, Beckman FJ, Conover WJ (2000) A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code. Tech. rep
9. Mehta P, Sait SM, Gürses D et al (2025) Aircraft wing rib component optimization using artificial neural network-assisted superb fairy-wren algorithm. Materialpruefung/Materials Test 67(9):1520–1527. https://doi.org/10.1515/mt-2025-0135
10. Mushtaq RT, Iqbal A, Wang Y et al (2023) Investigation and Optimization of Effects of 3D Printer Process Parameters on Performance Parameters. Materials 16(9). https://doi.org/10.3390/ma16093392
11. Oh Y, Sharp M, Sprock T et al (2021) Neural network-based build time estimation for additive manufacturing: A performance comparison. J Computat Design Eng 8(5):1243–1256. https://doi.org/10.1093/jcde/qwab044
12. Peng T, Sun W (2017) Energy modelling for FDM 3D printing from a life cycle perspective. Int J Manufactur Res 12(1):83–98. https://doi.org/10.1504/IJMR.2017.083651
13. Pradel P, Bibb R, Zhu Z et al (2017) Complexity Is Not For Free: The Impact of Component Complexity on Additive Manufacturing Build Time. Loughborough University. https://hdl.handle.net/2134/24338
14. Sait SM, Mehta P, Gürses D et al (2025a) Artificial neural network-assisted supercell thunderstorm algorithm for optimization of real-world engineering problems. Materialpruefung/Materials Test 67(9):1528–1536. https://doi.org/10.1515/mt-2025-0182
15. Sait SM, Mehta P, Ylldlz BS et al (2025b) Artificial neural network-infused polar fox algorithm for optimal design of vehicle suspension components. Materialpruefung/Materials Test 67(8):1400–1408. https://doi.org/10.1515/mt-2025-0043
16. Srikanth M, Mathew AT, Bhagchandani RK (2024) Model performance evaluation of build time using geometric shape complexity and process parameters in material extrusion. Addit Manuf 91. https://doi.org/10.1016/j.addma.2024.104337
17. Tang Y, Yang S, Zhao YF (2016) Sustainable design for additive manufacturing through functionality integration and part consolidation. In: Environmental Footprints and Eco-Design of Products and Processes. Springer, pp 101–144. https://doi.org/10.1007/978-981-10-0549-7_6
18. Trovato M, Belluomo L, Bici M et al (2025). Machine learning in design for additive manufacturing A state-of-the-art discussion for a support tool in product design lifecycle. https://doi.org/10.1007/s00170-025-15273-9
19. Tukey JW (1977) Exploratory data analysis. Addison-Wesley
20. Verma N, Boyer E, Verbeek J (2018) FeaStNet: Feature-Steered Graph Convolutions for 3D Shape Analysis. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition. IEEE Computer Society, pp 2598–2606. https://doi.org/10.1109/CVPR.2018.00275
21. Vidakis N, Kechagias JD, Petousis M et al (2023) The effects of FFF 3D printing parameters on energy consumption. Mater Manuf Processes 38(8):915–932. https://doi.org/10.1080/10426914.2022.2105882
22. Watanabe S (2025) Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance. arxiv:2304.11127
23. Wernet G, Bauer C, Steubing B et al (2016) The ecoinvent database version 3 (part I): overview and methodology. Int J Life Cycle Assess 21(9):1218–1230. https://doi.org/10.1007/s11367-016-1087-8

24. Williams G, Meisel NA, Simpson TW et al (2019) Design repository effectiveness for 3D convolutional neural networks: Application to additive manufacturing. J Mechan Design 141(11). https://doi.org/10.1115/1.4044199
25. Yildiz AR (2025) Comparison of crash performance of auxetic structures for CF15, PA6/GF30, and GF30PP materials. Materialpruefung/Materials Test 67(6):953–963. https://doi.org/10.1515/mt-2025-0118
26. Zhou Q, Jacobson A (2016) Thingi10K: A Dataset of 10,000 3D-Printing Models. Tech. rep. https://doi.org/10.48550/arXiv.1605.04797