AI-Driven Construction Intelligence

From Design Understanding to Manufacturing Optimization

Andrew Fisher, PhD

April. 30, 2026

OCRC Webinar Series

AI-Driven Construction Intelligence

From Design Understanding to Manufacturing Optimization

Andrew Fisher, PhD

April. 30, 2026

OCRC Webinar Series

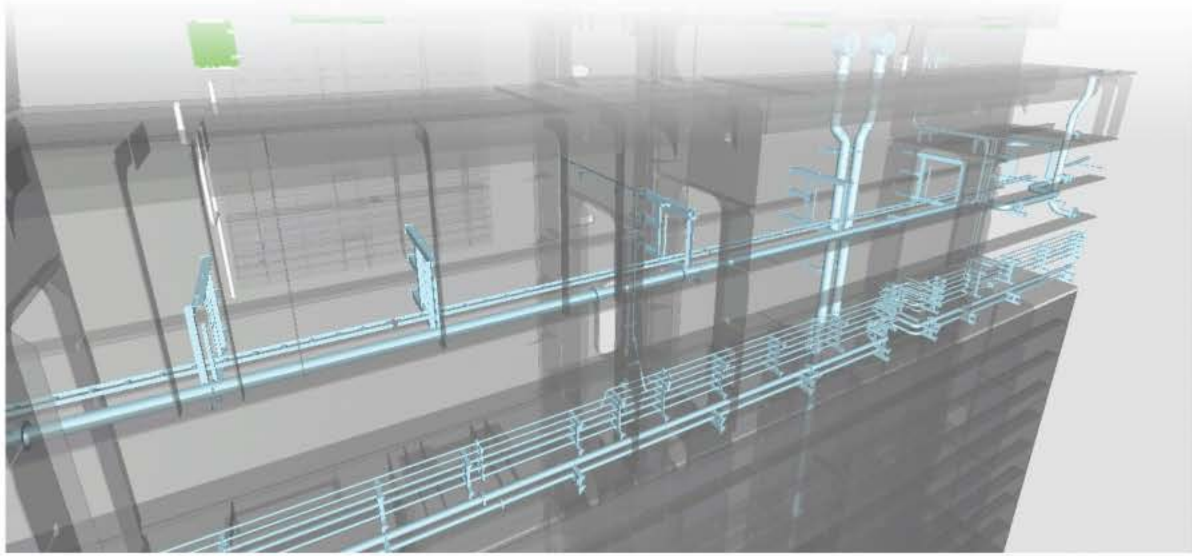
Transforming Construction with AI Integration

Andrew Fisher, PhD
AI-Driven Construction Intelligence

UNIVERSITY OF NEW BRUNSWICK

Transforming Construction with AI Integration

- Construction workflows are often split across separate stages such as **design interpretation, material estimation, and manufacturing planning** [1, 2, 3].
- AI offers an opportunity to **connect these stages more effectively**, helping information flow from early drawings to production decisions [4, 5, 6].
- The goal is to **reduce manual effort, improve consistency, shorten project timelines, and minimize material waste** [2, 7, 8].
- This presentation introduces three connected frameworks: **BIRD** for drawing-to-3D visualization, **JACK** for material estimation, and **PAAD** for manufacturing and panel layout optimization [9, 10, 11].
- This research, developed in collaboration with **Greenstone Building Products** in Brandon, Manitoba, focuses on making construction workflows more **efficient, explainable, and scalable**.



UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK

The Need for 3D Reconstruction

Early **3D visualization** is important for understanding client requirements, supporting planning, and improving cost estimation [1]:

- In practice, turning several **2D building drawings** into a usable 3D model is still a **time-intensive manual task** that depends heavily on expert interpretation [12].
- Existing AI methods for 3D reconstruction work well for **general images and scenes**, but they often struggle with architectural drawings because they were not designed for:
  - combining multiple building views consistently [13],
  - preserving exact geometric measurements rather than estimating depth [14], and
  - supporting real-world revisions and corrections [15].
- Methods based only on **floor plans** can recover layout information, but they usually cannot determine full building height, depth, or more complex features such as stacked openings [4].
- **BIRD** addresses these challenges through a **multi-view, noise-aware, geometry-preserving** framework for converting standard architectural drawings into 3D building representations [9].

Automating the Visualization Process

BIRD prepares building drawings for 3D reconstruction through three main steps:

- First, it detects the **important structural lines** in each drawing, such as walls, windows, and doors [16].
- Second, it uses a **dual-branch design** to separate meaningful building features from text, symbols, and other visual noise [17].
- Third, it **extracts measurements** from the drawing and uses them to **scale the detected lines accurately** before reconstruction [18].

- The model uses the same multi-output training objective as HED, allowing it to learn line features at several levels of detail simultaneously [20].
- This helps BIRD detect both broader structure and finer building details in architectural drawings.

Projecting Into a 3D Space

After detecting and scaling the line segments, BIRD maps each 2D drawing into a shared **3D coordinate system**:

- Each building view is reoriented so that its lines align with the **correct height, width, and depth axes**.
- The different perspectives are then **combined to form a single coherent 3D building envelope**.
- This wrapping process allows BIRD to handle **corners, angled walls, and irregular floor plans** more accurately than simpler projection methods.

BIRD: Building Image Reconstruction and Dimensioning

Engineering Applications of Artificial Intelligence 139 (2025) 109657

Contents lists available at ScienceDirect

ELSEVIER

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Building image reconstruction and dimensioning of the envelope from two-dimensional perspective drawings

Andrew Fisher [a,d,*], Lucas Moreira [b], Muntasir Billah [c], Pawan Lingras [a], Vijay Mago [d]

[a] Saint Mary's University, Halifax, Nova Scotia, Canada
[b] Lakehead University, Thunder Bay, Ontario, Canada
[c] University of Calgary, Calgary, Alberta, Canada
[d] York University, Toronto, Ontario, Canada

A R T I C L E I N F O

Keywords:
Convolutional neural networks
Line segment detection
Text extraction
Building image reconstruction
Three-dimensional modeling

A B S T R A C T

In the construction industry, a project typically begins with the creation of two-dimensional (2D) building plans, defining the client's specifications. Using these plans, a digital three-dimensional (3D) model is developed to visualize the anticipated outcome and to verify the model's alignment with the client's expectations. The process of converting from 2D to 3D can become time-intensive if there is a need for modifications or if the project's overall complexity is high. To enhance efficiency and accuracy, this research introduces an end-to-end framework referred to as *BIRD* which stands for Building Image Reconstruction and Dimensioning. *BIRD* is capable of accepting five 2D perspective drawings of a building as inputs and generating a proportionate 3D model of the building envelope as an output. This is accomplished through the integration of multiple techniques that use convolutional neural networks to extract a refined set of line segments, identify measurements, and align each perspective with the floor plan drawing. The key contributions of this study includes: (1) a novel deep learning model designed for the identification of line segments in building plans; (2) novel algorithms that facilitate the generation of information required for 3D modeling; (3) an end-to-end framework for building reconstruction; and (4) novel performance metrics specifically tailored for the 2D to 3D conversion challenge. The practical application of this research was validated through the use of complete building plans provided by an industry partner. In summary, it was observed that *BIRD* demonstrated high accuracy in the creation of 3D visualizations, highlighting its real-world efficacy.

The Need for 3D Reconstruction

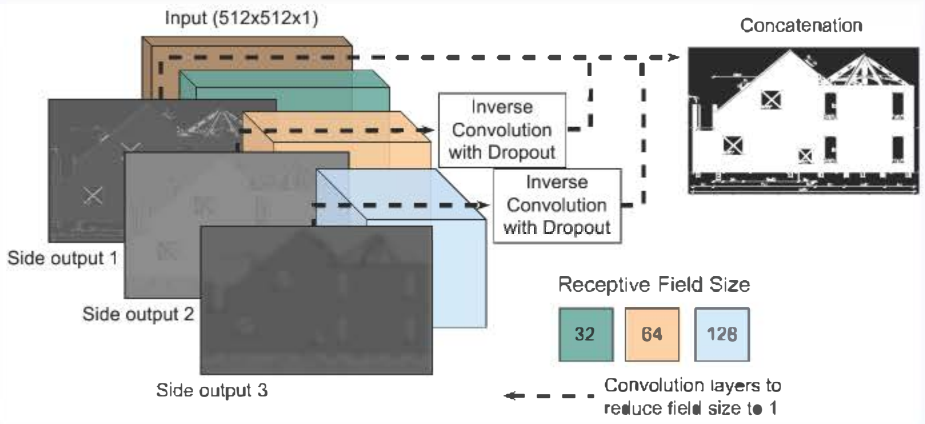
Early **3D visualization** is important for understanding client requirements, supporting planning, and improving cost estimation [1]:

- In practice, turning several **2D building drawings** into a usable 3D model is still a **time-intensive manual task** that depends heavily on expert interpretation [12].
- Existing AI methods for 3D reconstruction work well for **general images and scenes**, but they often struggle with architectural drawings because they were not designed for:
  - combining multiple building views consistently [13],
  - preserving exact geometric measurements rather than estimating depth [14], and
  - supporting real-world revisions and corrections [15].
- Methods based only on **floor plans** can recover layout information, but they usually cannot determine full building height, depth, or more complex features such as stacked openings [4].
- **BIRD** addresses these challenges through a **multi-view, noise-aware, geometry-preserving** framework for converting standard architectural drawings into 3D building representations [9].

Automating the Visualization Process

BIRD prepares building drawings for 3D reconstruction through three main steps:

- First, it detects the **important structural lines** in each drawing, such as walls, windows, and doors [16].
- Second, it uses a **dual-branch design** to separate meaningful building features from text, symbols, and other visual noise [17].
- Third, it extracts measurements from the drawing and uses them to **scale the detected lines accurately** before reconstruction [18].



- The model uses the same multi-output training objective as HED, allowing it to learn line features at several levels of detail simultaneously [16].
- This helps BIRD detect both broader structure and finer building details in architectural drawings.

Adapting HED for Building Plans

BIRD builds on an exisiting approach for line detection called HED, but modifies it to better suit architectural drawings.

Learn to resize outputs to match the original drawing

Transpose convolutions are used so detected features can be mapped back to the original image size more accurately [19].

Simplify the input representation

The model works on **black-and-white drawings**, which reduces complexity and improves training efficiency.

Reduce unnecessary model complexity

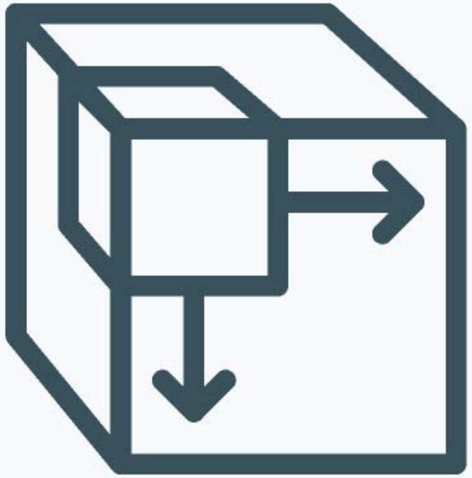
Dropout layers replace later output blocks to reduce the number of trainable parameters while preserving key line information [22].

Learn to resize outputs to match the original drawing

Transpose convolutions are used so detected features can be mapped back to the original image size more accurately [19].

Simplify the input representation

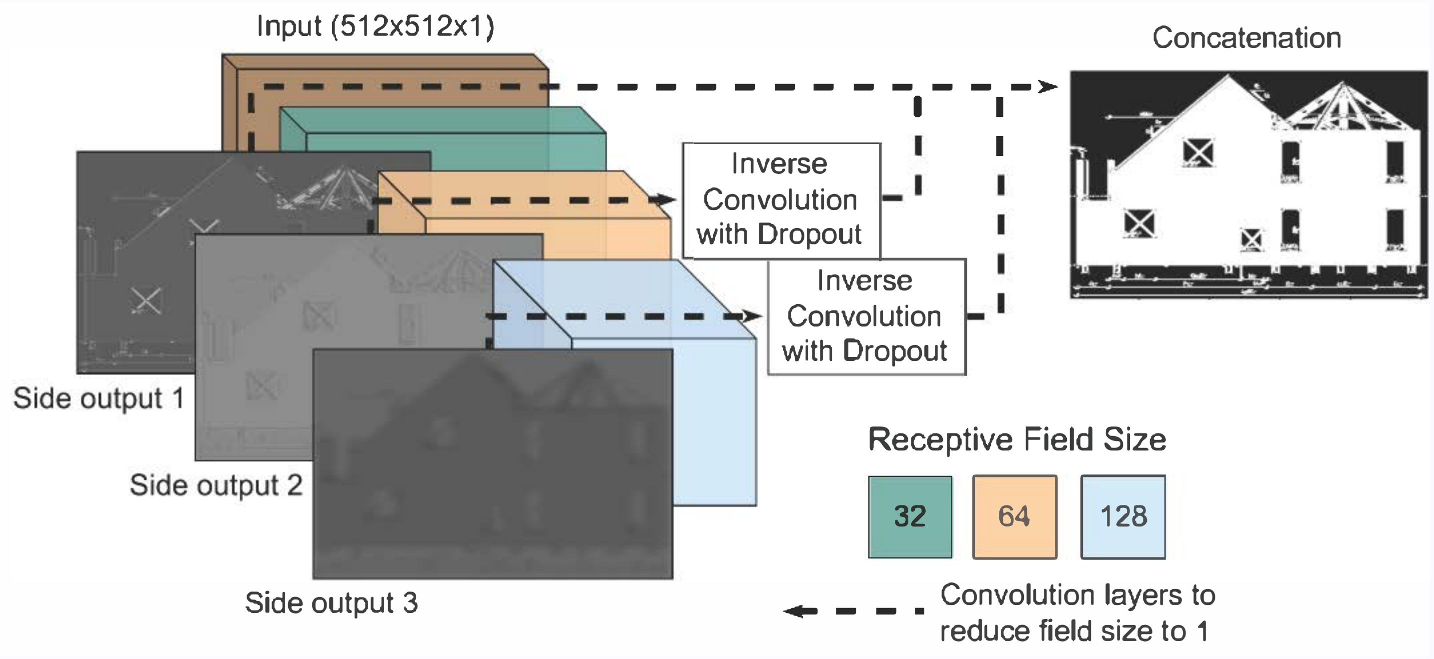
The model works on **black-and-white drawings**, which reduces complexity and improves training efficiency.



Reduce unnecessary model complexity

Dropout layers replace later output blocks to reduce the number of trainable parameters while preserving key line information [22].

ts from the drawing and uses them to **scale the detected lines**
ion [18].

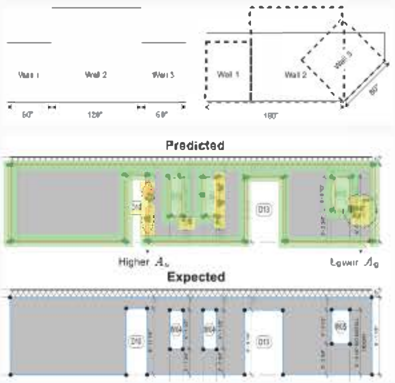


- The model uses the same multi-output training objective as HED, allowing it to learn line features at several levels of detail simultaneously [16].
- This helps BIRD detect both broader structure and finer building details in architectural drawings.

Projecting Into a 3D Space

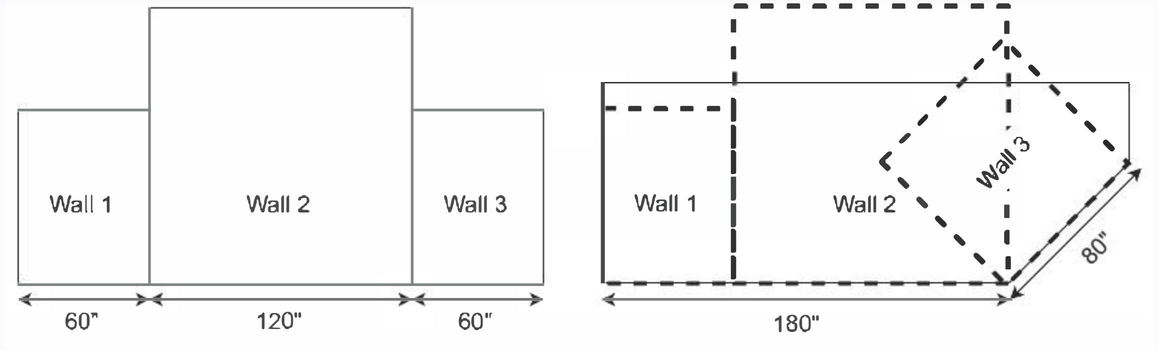
After detecting and scaling the line segments, BIRD maps each 2D drawing into a shared **3D coordinate system**:

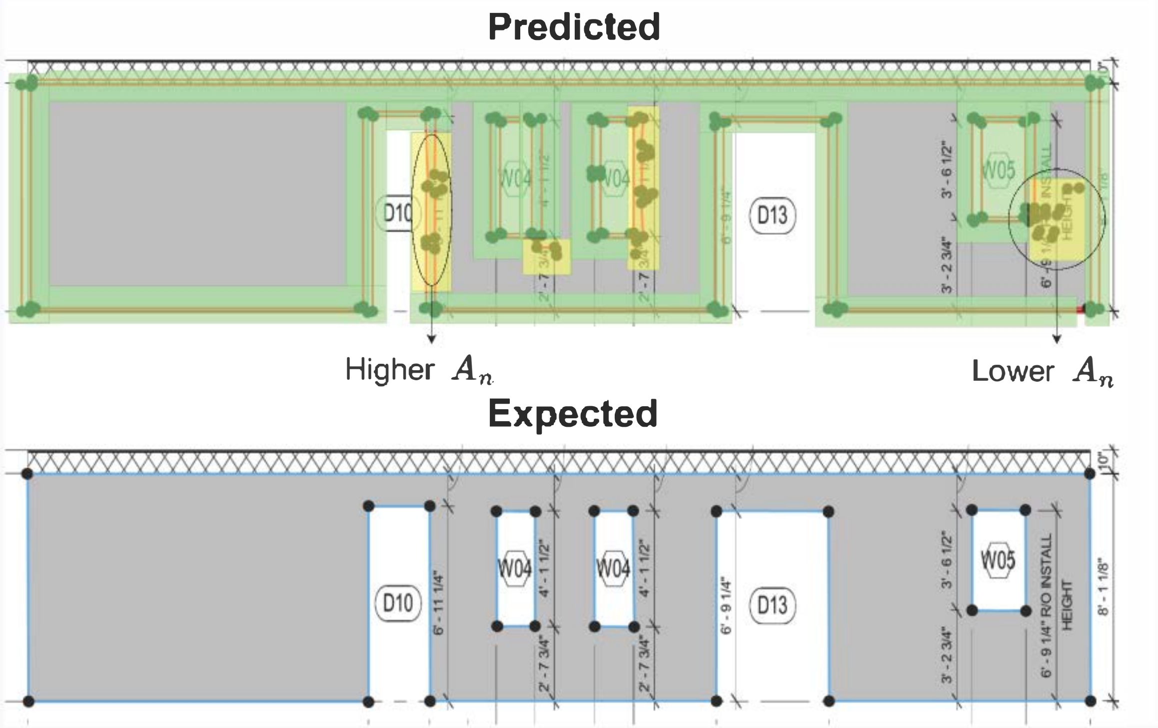
- Each building view is reoriented so that its lines align with the correct **height, width, and depth axes**.
- The different perspectives are then combined to form a single **coherent 3D building envelope**.
- This wrapping process allows BIRD to handle **corners, angled walls, and irregular floor plans** more accurately than simpler projection methods.



- Match predicted line segments to their closest ground-truth lines.
- Use average distance to measure **closest-line accuracy**.
- Identify unmatched predicted lines to measure **noise impact and extra-line accuracy**.
- Count the total number of extra lines (N) to quantify visual clutter.

$$L_c = \{\hat{p} \mid p \in P_i \wedge \hat{p} \in \hat{P}_i, \mathrm{dist}(p, \hat{p}) < \mathrm{closest\_dist}(p)\}$$
$$A_c = 100 - avg\_norm\_dist(L_c)$$
$$L_n = \{p \mid \hat{p} \in \hat{P}_i \wedge p \in P_i, \hat{p} \notin L_c \wedge (dist(\hat{p}, p) < closest\_dist(\hat{p}))\}$$
$$A_n = 100 - avg\_norm\_dist(L_n), \; N = count(L_n)$$





- Match predicted line segments to their closest ground-truth lines.
- Use average distance to measure **closest-line accuracy**.
- Identify unmatched predicted lines to measure **noise impact and extra-line accuracy**.
- Count the total number of extra lines (N) to quantify visual clutter.

$$L_c = \{\hat{p} \mid p \in P_i \wedge \hat{p} \in \hat{P}_i, \mathrm{dist}(p, \hat{p}) < \mathrm{closest\_dist}(p)\}$$
$$A_c = 100 - avg\_norm\_dist(L_c)$$
$$L_n = \{p \mid \hat{p} \in \hat{P}_i \wedge p \in P_i, \hat{p} \notin L_c \wedge (dist(\hat{p}, p) < closest\_dist(\hat{p}))\}$$
$$A_n = 100 - avg\_norm\_dist(L_n), N = count(L_n)$$

nportant for understanding client requirements, supporting planning, tion [1]:

- eral **2D building drawings** into a usable 3D model is still a **time-** k that depends heavily on expert interpretation [12].
- 3D reconstruction work well for **general images and scenes**, but they hitectural drawings because they were not designed for:
  - e building views consistently [13],
  - ometric measurements rather than estimating depth [14], and
  - rld revisions and corrections [15].
- **floor plans** can recover layout information, but they usually cannot height, depth, or more complex features such as stacked openings [4].

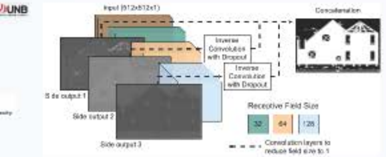
challenges through a **multi-view, noise-aware, geometry-preserving** ng standard architectural drawings into 3D building representations [9].

UNB
UNIVERSITY OF NEW BRUNSWICK

the Visualization Process

rawings for 3D reconstruction through three main steps:

- **tant structural lines** in each drawing, such as walls, windows, and doors
- **anch design** to separate meaningful building features from text, symbols,].
- ments from the drawing and uses them to **scale the detected lines** truction [18].



- The model uses the same multi-output training objective as HED, allowing it to learn line features at several levels of detail simultaneously [16].
- This helps BIRD detect both broader structure and finer building details in architectural drawings.

UNB
UNIVERSITY OF NEW BRUNSWICK

nto a 3D Space

he line segments, BIRD maps each 2D drawing into a shared **3D**

eoriented so that its lines align with the correct **height, width, and depth**

ves are then combined to form a single **coherent 3D building envelope**. allows BIRD to handle **corners, angled walls, and irregular floor plans** impler projection methods.

BIRD: Bu Image Reconstru and Dime

Engineering Applications of Artificial Inte

Contents lists available at

Engineering Applications of

journal homepage: www.elsevie

Building image reconstruction and dimensioning two-dimensional perspective drawings

Andrew Fisher [a,d,*], Lucas Moreira [b], Muntasir Billah [c], Pawan

[a] Saint Mary's University, Halifax, Nova Scotia, Canada
[b] Lakehead University, Thunder Bay, Ontario, Canada
[c] University of Calgary, Calgary, Alberta, Canada
[d] York University, Toronto, Ontario, Canada

ARTICLE INFO

Keywords:
Convolutional neural networks
Line segment detection
Text extraction
Building image reconstruction
Three-dimensional modeling

ABSTRACT

In the construction industry, a proj plans, defining the client's specificati to visualize the anticipated outcom process of converting from 2D to 3 the project's overall complexity is h to-end framework referred to as B BIRD is capable of accepting five

UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK


Expert Systems

WILEY

ORIGINAL ARTICLE

Jointly Trained Automation of Explainable Construction Material Knowledge

ABSTRACT

JACK: Jointly Trained Automation of Explainable Construction Material Knowledge

The Potential of Estimation Automation

Accurate material estimation is essential for controlling costs, protecting profit margins, and maintaining client trust in construction projects:

- In practice, estimation is still often based on **manual spreadsheets and expert judgment**, which can be slow, inconsistent, and difficult to scale [5].
- Traditional **BIM-based and rule-driven systems** improve organization, but they still rely heavily on predefined formulas and human input [21].
- Earlier machine learning approaches have mainly focused on **cost prediction**, but many of them:
  - require **large, high-quality datasets** [22],
  - **predict costs rather than material quantities** [23], and
  - offer limited **explainability**, which can reduce industry trust [24].
- JACK addresses these limitations through **an explainable, cascaded neural network framework** that learns relationships between materials, works effectively with limited data, and produces practical quantity estimates for real construction workflows.

Leveraging Joint Training Techniques

- Many building materials depend on the same project information, such as **height, length, floor area, and openings.**
- Instead of training a separate model for every material in isolation, JACK **trains connected networks using shared inputs** [25].
- This allows the model to learn common building features once and reuse them across multiple material predictions.
- The result is **more efficient training, more consistent estimates, and a framework that can scale to new materials more easily.**

$$h_{1,n,i} = \Phi(W_{1,n,i}^{T} x_{n,i})$$

where $h_{1,n,i}$ = nodes in first hidden layer of network n, connected to input node i, Φ = activation function, $W_{1,n,i}^{T}$ = corresponding set of weights, and $x_{n,i}$ = input value

- This equation formalizes how each material network draws from a shared pool of **project parameters**.
- In practice, that means materials influenced by the same building features can learn from **a common representation** rather than repeating the same learning separately.
- This reduces redundancy and helps keep **predictions aligned across materials.**

Cascaded Training Across Networks

- Some material estimates depend on others, so the networks cannot always be treated as independent.
- In JACK, an **upstream ("macro") network** can provide an output that becomes an input to a **dependent ("micro") network** [26].
- During training, the dependent network's error is allowed to **flow back through this connection**, so both networks learn from the relationship.
- This helps JACK capture **real material dependencies**, improving accuracy and generalization even when data is limited.

$$\frac{\partial l_i}{\partial w_{(h_n, I_n)}} = \delta(I_n, I_n) \cdot h_n$$

1. The dependent micro network uses the macro network's output as one of its inputs, creating a link through which error signals can be shared during backpropagation.

2. This expanded gradient then accounts for all paths through the micro network.

$$\frac{\partial l_i}{\partial w_{(h_n, h_n)}} = \delta(h_n, h_n) \cdot h_i$$

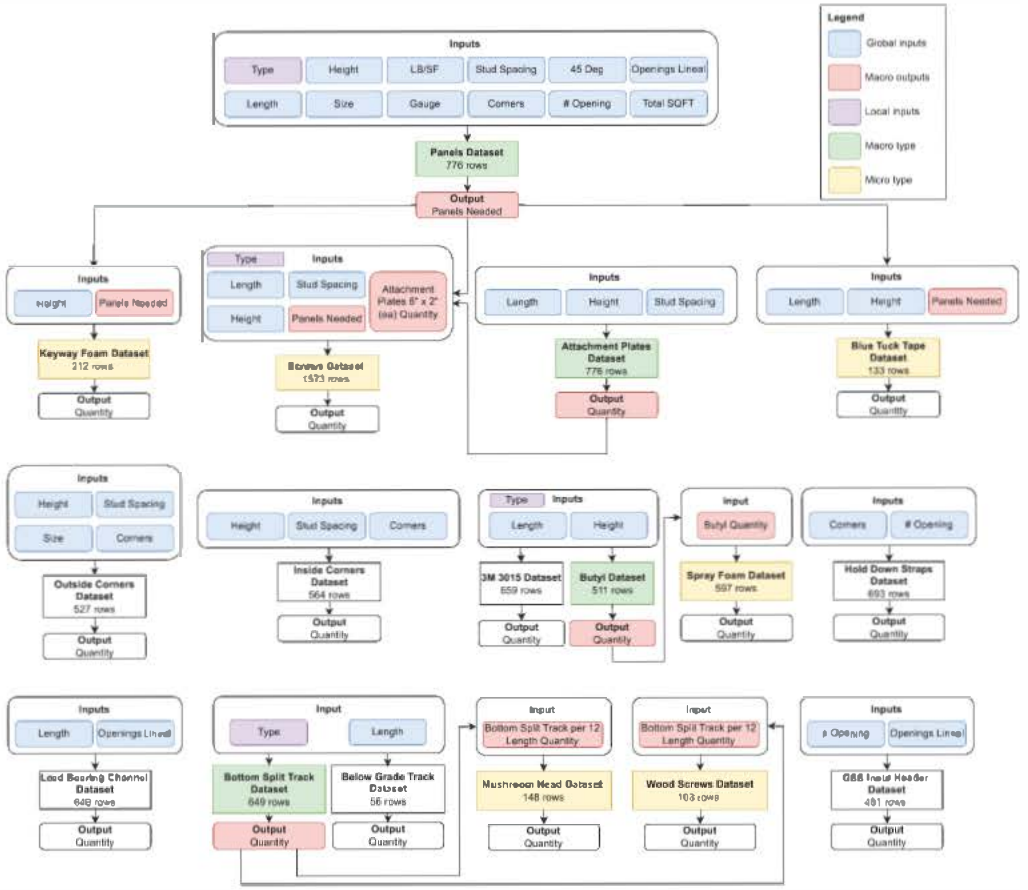
3. The macro network then continues with standard backpropagation, updating its own weights while incorporating feedback coming from the connected micro network.

Initializing the JACK Framework

- JACK represents each **material type** as its own neural network, using the project features most relevant to that material [10].
- Some of these **networks** are connected, because one material estimate may depend on another.
- For example, **panels act as a macro network and screws act as a dependent micro network**, allowing information to flow between related material estimates.
- During training, weights are transferred across connected network pairs, and parts of the macro network are later **frozen** to stabilize what it has already learned while refining the remaining layers.

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

- The framework is evaluated using mean absolute error (MAE), which measures the average difference between predicted and actual material quantities.
- This makes the results easy to interpret in practice: **smaller errors mean more reliable estimates and lower financial risk.**



The Potential of Estimation Automation

Accurate **material estimation** is essential for controlling costs, protecting profit margins, and maintaining client trust in construction projects:

- In practice, estimation is still often based on **manual spreadsheets and expert judgment**, which can be slow, inconsistent, and difficult to scale [5].
- Traditional **BIM-based and rule-driven systems** improve organization, but they still rely heavily on predefined formulas and human input [21].
- Earlier machine learning approaches have mainly focused on **cost prediction**, but many of them:
  - require **large, high-quality datasets** [22],
  - predict **costs rather than material quantities** [23], and
  - offer limited **explainability**, which can reduce industry trust [24].
- **JACK** addresses these limitations through an **explainable, cascaded neural network framework** that learns relationships between materials, works effectively with limited data, and produces practical quantity estimates for real construction workflows.

UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK

Leveraging Joint Training Techniques

- Many building materials depend on the same project information, such as **height, length, floor area, and openings**.
- Instead of training a separate model for every material in isolation, JACK **trains connected networks using shared inputs** [25].
- This allows the model to learn common building features once and reuse them across multiple material predictions.
- The result is **more efficient training, more consistent estimates, and a framework that can scale to new materials more easily.**

$$\bar{h}_{1,n,i} = \Phi(W_{1,n,i}^{T}\bar{x}_{n,i})$$

where $\bar{h}_{1,n,i}$ = nodes in first hidden layer of network *n*, connected to input node *i*, Φ = activation function, $W_{1,n,i}^{T}$ = transposed set of weights, and $\bar{x}_{n,i}$ = input value

- This equation formalizes how each material network draws from a **shared pool of project parameters**.
- In practice, that means materials influenced by the same building features can learn from a **common representation** rather than repeating the same learning separately.
- This reduces redundancy and helps keep predictions **aligned across materials**.

efficient training, more consistent estimates, and a framewo more easily.

$$\bar{h}_{1,n,i} = \Phi(W^T_{1,n,i}\bar{x}_{n,i})$$

where $\bar{h}_{1,n,i}$ = nodes in first hidden layer of network *n*, connected to input node *i*, Φ = activation function, $W^T_{1,n,i}$ = transposed set of weights, and $\bar{x}_{n,i}$ = input value

- This equation formalizes how each material network draws from a **shared pool of project parameters**.
- In practice, that means materials influenced by the same building features can learn from a **common representation** rather than repeating the same learning separately.
- This reduces redundancy and helps keep predictions **aligned across materials**.

UNB
UNIVERSITY OF NEW BRUNSWICK
EST. 1785

Cascaded Training Across Networks

- Some material estimates depend on others, so the networks cannot always be treated as independent.
- In JACK, an **upstream ("macro") network** can provide an output that becomes an input to a **dependent ("micro") network** [26].
- During training, the dependent network's error is allowed to **flow back through this connection**, so both networks learn from the relationship.
- This helps JACK capture **real material dependencies**, improving accuracy and generalization even when data is limited.

$$\frac{\partial l_l}{\partial w_{(h_o,l_n)}} = \delta(l_n, l_\bullet) \cdot h_\bullet$$

1. The dependent micro network (l) uses the macro network's (h) output as one of its inputs, creating a link through which error signals can be shared during backpropagation (w).

$$\delta(l_n, l_o) = \frac{\partial l_l}{\partial l_o} \cdot \Phi'(l_{o_a}) \cdot \left[\sum_{[l_n, l_{n-1}, \ldots, l_k, l_o] \in P} \frac{\partial l_{o_a}}{\partial l_{k_a}} \prod_{x=n}^{k-1} \frac{\partial l_{x+1_a}}{l_{x_a}} \right]$$

2. This expanded gradient term accounts for all valid paths (P) through the micro network, ensuring that error signals are propagated consistently through every connected route.

$$\frac{\partial h_l}{\partial w_{(h_i,h_n)}} = \delta(h_n, h_o) \cdot h_i$$

3. The macro network then continues with standard backpropagation, updating its own weights while incorporating feedback coming from the connected micro network.

$$\frac{\partial l_l}{\partial w_{(h_o, l_n)}} = \delta(l_n, l_o) \cdot h_o$$

1. The dependent micro network (*l*) uses the macro network's (*h*) output as one of its inputs, creating a link through which error signals can be shared during backpropagation (*w*).

$$\delta(l_n, l_o) = \frac{\partial l_l}{\partial l_o} \cdot \Phi'(l_{o_a}) \cdot \left[\sum_{[l_n, l_{n-1}, \ldots, l_k, l_o] \in P} \frac{\partial l_{o_a}}{\partial l_{k_a}} \prod_{x=n}^{k-1} \frac{\partial l_{x+1_a}}{l_{x_a}} \right]$$

2. This expanded gradient term accounts for all valid paths (*P*) through the micro network, ensuring that error signals are propagated consistently through every connected route.

$$\frac{\partial h_l}{\partial w_{(h_i, h_n)}} = \delta(h_n, h_o) \cdot h_i$$

3. The macro network then continues with standard backpropagation, updating its own weights while incorporating feedback coming from the connected micro network.

Initializing the JACK Framework

- JACK represents each **material type** as its own neural network, using the project features most relevant to that material [10].
- Some of these **networks** are connected, because one material estimate may depend on another.
- For example, **panels act as a macro network** and **screws act as a dependent micro network**, allowing information to flow between related material estimates.
- During training, weights are transferred across connected network pairs, and parts of the macro network are later **frozen** to stabilize what it has already learned while refining the remaining layers.

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

- The framework is evaluated using **mean absolute error (MAE)**, which measures the average difference between predicted and actual material quantities.
- This makes the results easy to interpret in practice: **smaller errors mean more reliable estimates and lower financial risk.**

e, **panels act as a macro network** and **screws act as a dependent micro netw**
n to flow between related material estimates.
ning, weights are transferred across connected network pairs, and parts of the m
ozen to stabilize what it has already learned while refining the remaining layers.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$$

- The framework is evaluated using **mean absolute error (MAE)**, which measures the average difference between predicted and actual material quantities.
- This makes the results easy to interpret in practice: **smaller errors mean more reliable estimates and lower financial risk.**

echniques

n, such as **height, length, floor area,**

on, JACK **trains connected networks**

nd reuse them across multiple

s, and a framework that can scale

es how each material
shared pool of

s materials influenced
eatures can learn from
tion rather than
aning separately.
cy and helps keep
ross materials.

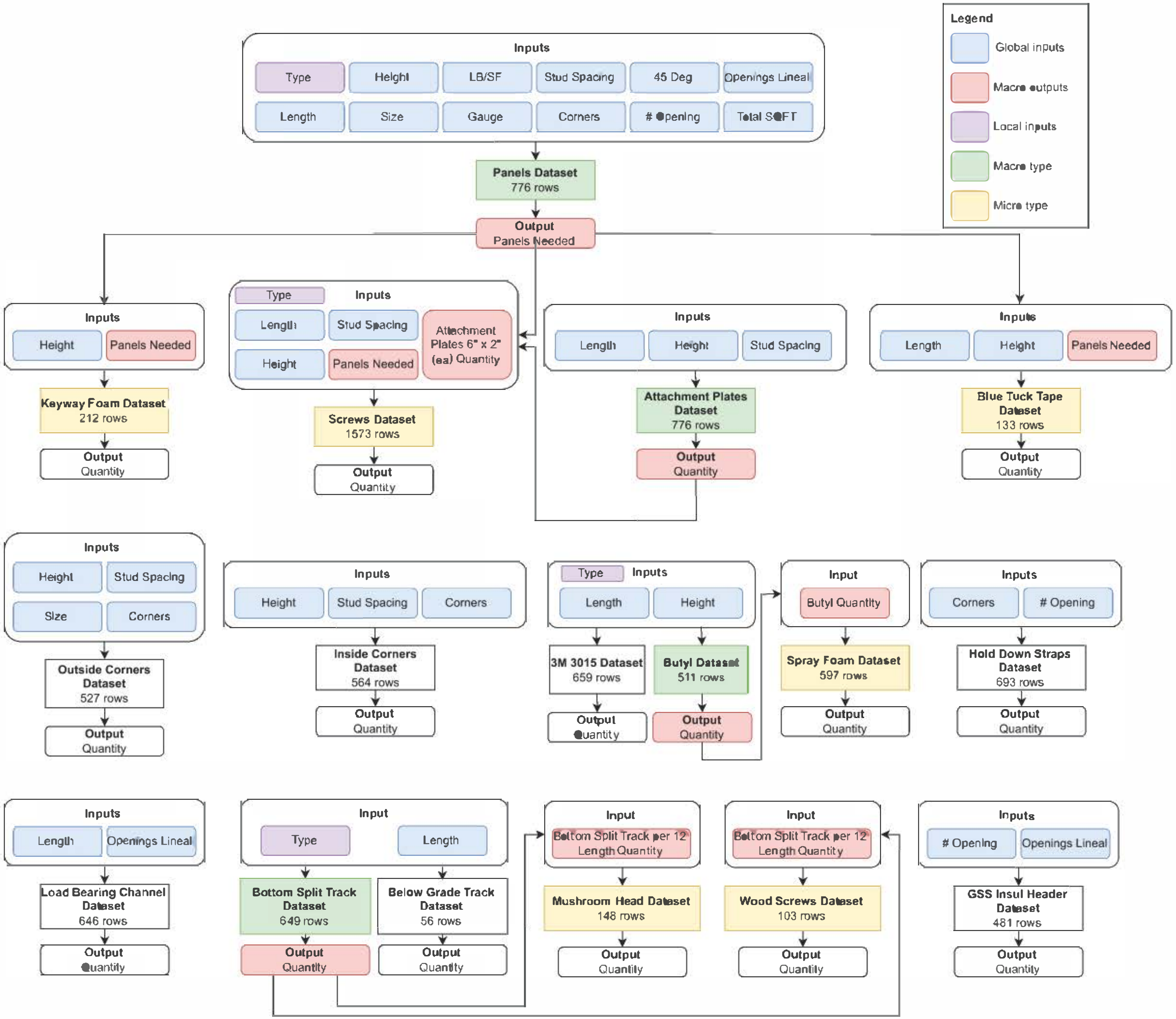
ork

using the project features most relevant

stimate may depend on another.
a dependent micro network, allowing

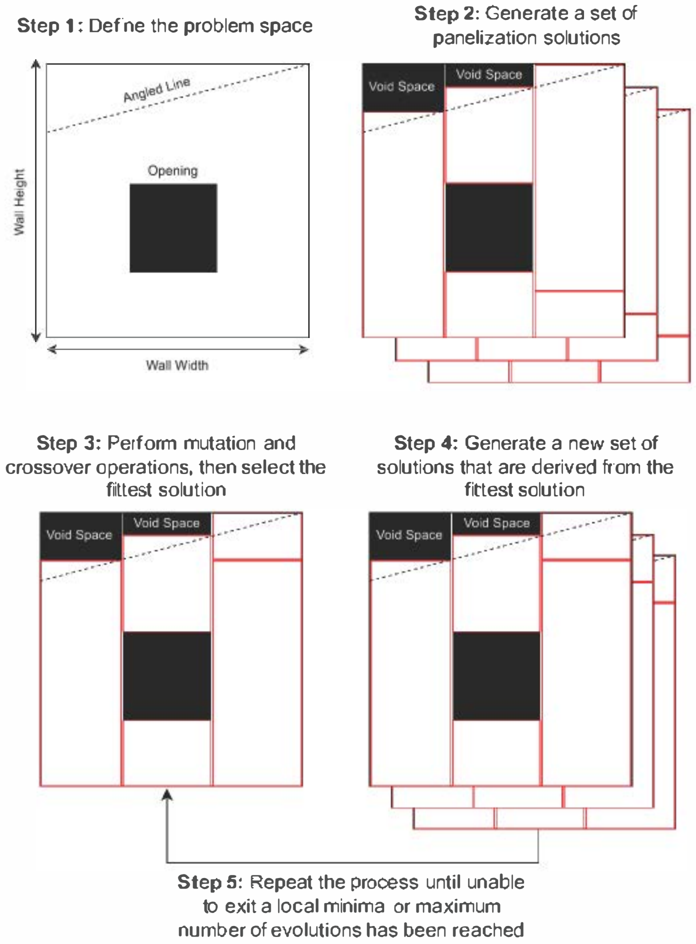
rk pairs, and parts of the macro network
ning the remaining layers.

uated using **mean absolute**
easures the average difference
d actual material quantities.
easy to interpret in practice:
more reliable estimates and



UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK

PAAD: Panelization Algorithm for Architectural Designs



PLOS ONE

RESEARCH ARTICLE

PAAD: Panelization algorithm for architectural designs

Andrew Fisher[1], Xing Tan[2], Muntasir Billah[3], Pawan Lingras[1], Jimmy Huang[4], Vijay Mago[5]*

1 Department of Mathematics and Computing Science, Saint Mary's University, Halifax, Nova Scotia, Canada, 2 Department of Computer Science, Lakehead University, Thunder Bay, Ontario, Canada, 3 Department of Civil Engineering, University of Calgary, Calgary, Alberta, Canada, 4 School of Information Technology, York University, Toronto, Ontario, Canada, 5 School of Health Policy and Management, York University, Toronto, Ontario, Canada

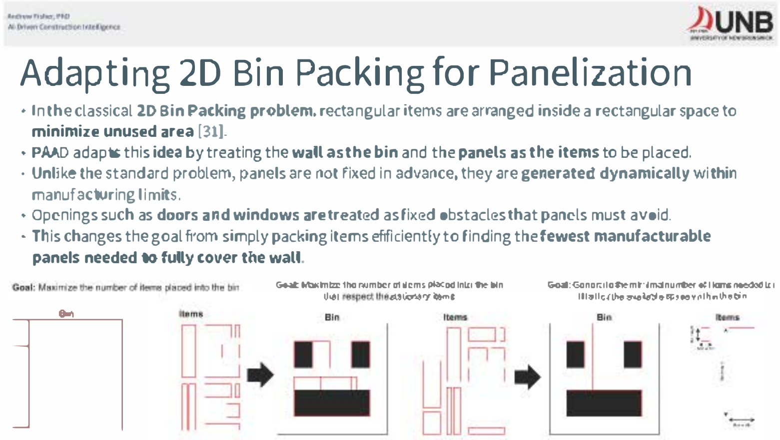
* vmago@yorku.ca

Check for updates

Manufacturing Optimization for Building Panels

Efficient **panelization** is important for reducing cost, minimizing material waste, and keeping manufacturing plans practical in construction:

- Current **rule-based and BIM-driven systems** can automate layout generation, but they are **often rigid** and require extensive project-specific rules [27].
- Classical **optimization methods** such as **Integer Programming** and the **Cutting Stock Problem** can improve material usage, but they become harder to apply when wall shapes **are** irregular or highly constrained [28, 29].
- More recent approaches using **AR or human-guided heuristics** can improve interpretability and quality control, but they also add procedural complexity [30].
- **PAAD** addresses these challenges by treating panelization as a modified **2D Bin Packing** problem and solving it with a **genetic evolutionary algorithm** that can handle manufacturing limits, angled walls, and multiple openings while still producing practical, interpretable layouts [6, 31].



Adapting 2D Bin Packing for Panelization

- In the classical **2D Bin Packing problem**, rectangular items are arranged inside a rectangular space to **minimize unused area** [31].
- PAAD adapts this **idea** by treating the **wall as the bin** and the **panels as the items** to be placed.
- Unlike the standard problem, panels are not fixed in advance, they are **generated dynamically** within manufacturing limits.
- Openings such as **doors and windows** are treated as fixed obstacles that panels must avoid.
- This changes the goal from simply packing items efficiently to finding the **fewest manufacturable panels needed to fully cover the wall**.

Goal: Maximize the number of items placed into the bin

Bin

Items

Goal: Maximize the number of items placed into the bin that respect the stationary items

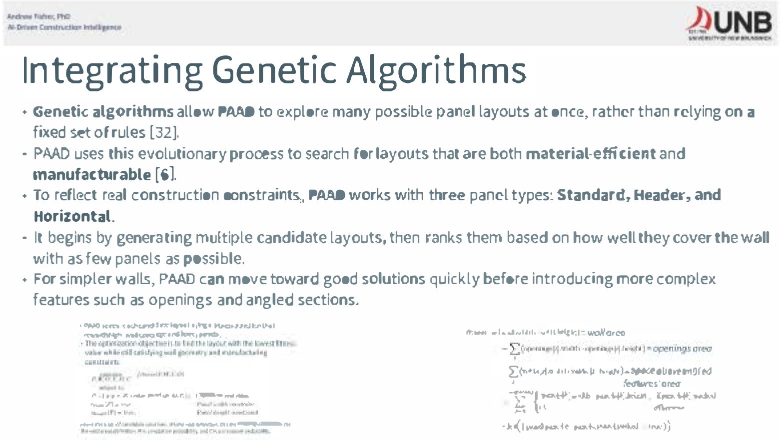
Bin

Items

Goal: Generate the minimal number of items needed to fill the available space in the bin

Bin

Items



Integrating Genetic Algorithms

- **Genetic algorithms** allow **PAAD** to explore many possible panel layouts at once, rather than relying on a fixed set of rules [32].
- PAAD uses this evolutionary process to search for layouts that are both **material-efficient** and **manufacturable** [6].
- To reflect real construction constraints, **PAAD** works with three panel types: **Standard, Header, and Horizontal**.
- It begins by generating multiple candidate layouts, then ranks them based on how well they cover the wall with as few panels as possible.
- For simpler walls, PAAD can move toward good solutions quickly before introducing more complex features such as openings and angled sections.

Manufacturing Optimization for Building Panels

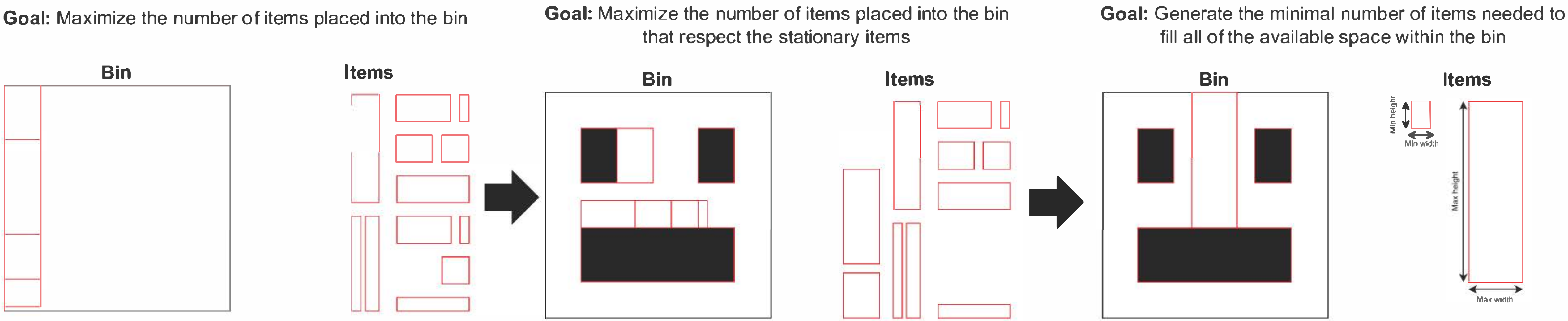
Efficient **panelization** is important for reducing cost, minimizing material waste, and keeping manufacturing plans practical in construction:

- Current **rule-based and BIM-driven systems** can automate layout generation, but they are often **rigid** and require extensive project-specific rules [27].
- Classical optimization methods such as **Integer Programming** and the **Cutting Stock Problem** can improve material usage, but they become harder to apply when wall shapes are irregular or highly constrained [28, 29].
- More recent approaches using **AR or human-guided heuristics** can improve interpretability and quality control, but they also add procedural complexity [30].
- **PAAD** addresses these challenges by treating panelization as a modified **2D Bin Packing** problem and solving it with a **genetic evolutionary algorithm** that can handle manufacturing limits, angled walls, and multiple openings while still producing practical, interpretable layouts [6, 31].

UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK

Adapting 2D Bin Packing for Panelization

- In the classical **2D Bin Packing problem**, rectangular items are arranged inside a rectangular space to **minimize unused area** [31].
- PAAD adapts this idea by treating the **wall as the bin** and the **panels as the items** to be placed.
- Unlike the standard problem, panels are not fixed in advance, they are **generated dynamically** within manufacturing limits.
- Openings such as **doors and windows** are treated as fixed obstacles that panels must avoid.
- This changes the goal from simply packing items efficiently to finding the **fewest manufacturable panels needed to fully cover the wall**.



UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK

Integrating Genetic Algorithms

- **Genetic algorithms** allow PAAD to explore many possible panel layouts at once, rather than relying on a fixed set of rules [32].
- PAAD uses this evolutionary process to search for layouts that are both **material-efficient** and **manufacturable** [6].
- To reflect real construction constraints, PAAD works with three panel types: **Standard, Header, and Horizontal**.
- It begins by generating multiple candidate layouts, then ranks them based on how well they cover the wall with as few panels as possible.
- For simpler walls, PAAD can move toward good solutions quickly before introducing more complex features such as openings and angled sections.

- PAAD scores each candidate layout using a fitness function that rewards high wall coverage and fewer panels.
- The optimization objective is to find the layout with the lowest fitness value while still satisfying wall geometry and manufacturing constraints

$$
\begin{aligned}
\underset{P, W, O, V, M, C}{\text{minimize}} \quad & \mathit{fitness}(P, W, V, O) \\
\text{subject to} \quad & \\
& P = \{p \mid p \in P, \mathit{evolve\_panels}(p, M, C)\}, && \textit{Population evolution,} \\
& c_{width}(P) = \mathit{true}, && \textit{Panel width constraint,} \\
& c_{height}(P) = \mathit{true}, && \textit{Panel height constraint}
\end{aligned}
$$

where *P* is a set of candidate solutions, *W* is the wall definition, *O* is the openings definition, *V* is the void areas definition, *M* is a mutation probability, and *C* is a crossover probability.

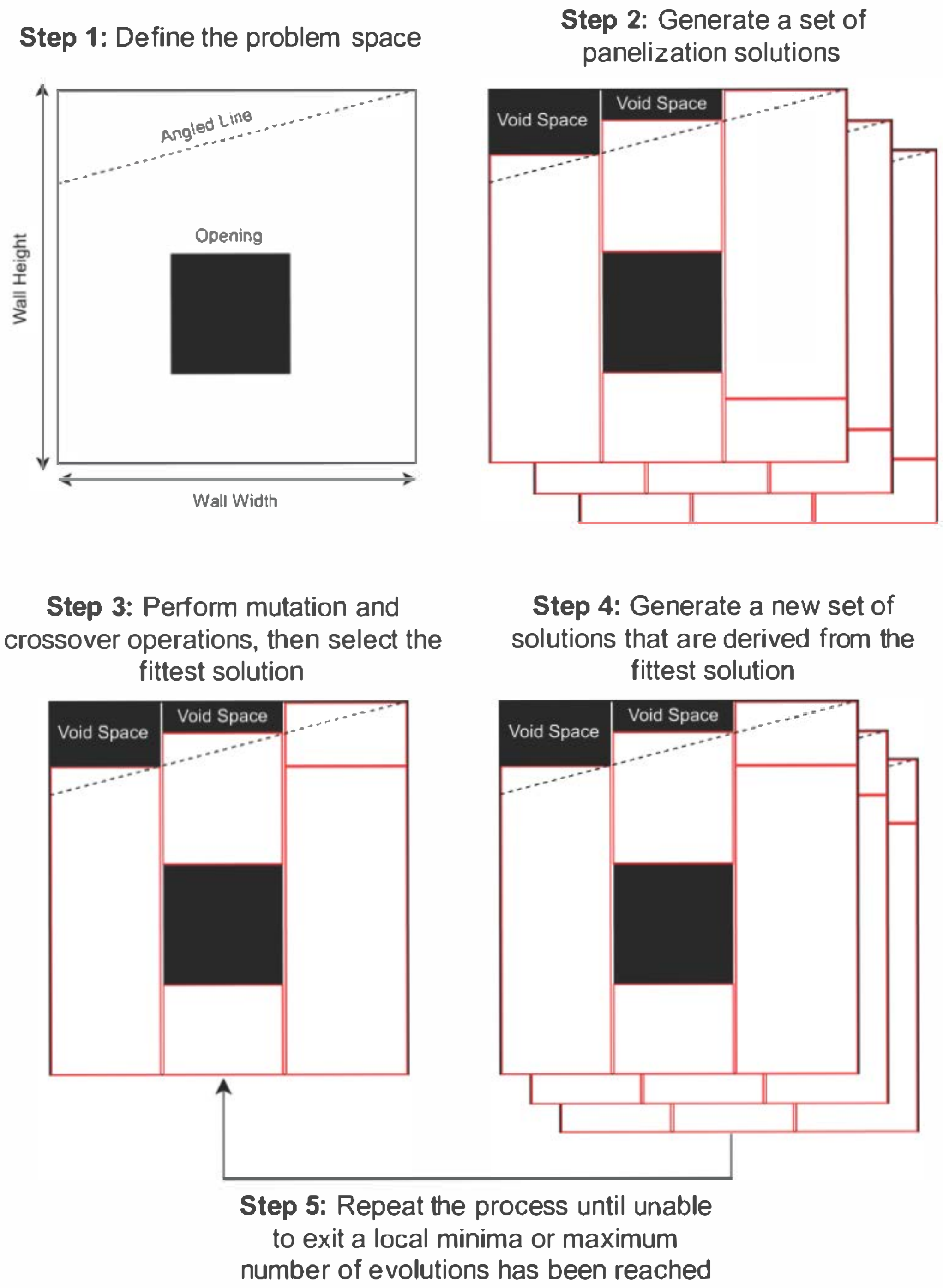
$$
\begin{aligned}
\text{fitness} = {} & (\text{wall.width} \cdot \text{wall.height}) = \textit{wall area} \\
& - \sum_i (\text{openings}[i].\text{width} \cdot \text{openings}[i].\text{height}) = \textit{openings area} \\
& - \sum_j (\text{voids}[j].\text{width} \cdot \text{voids}[j].\text{height}) = \textit{space above angled features' area} \\
& - \sum_{k=0}^{\text{len(panels)}} \begin{cases} \text{panels}[k].\text{width} \cdot \text{panels}[k].\text{height}, & \text{if panels}[k].\text{packed} \\ 0, & \text{otherwise} \end{cases} \\
& + \text{len}\Big(\{\text{panel} \mid \text{panel} \in \text{panels}, \text{panel.packed} = \text{true}\}\Big)
\end{aligned}
$$

- PAAD scores each candidate layout using a fitness function that rewards high wall coverage and fewer panels.
- The optimization objective is to find the layout with the lowest fitness value while still satisfying wall geometry and manufacturing constraints

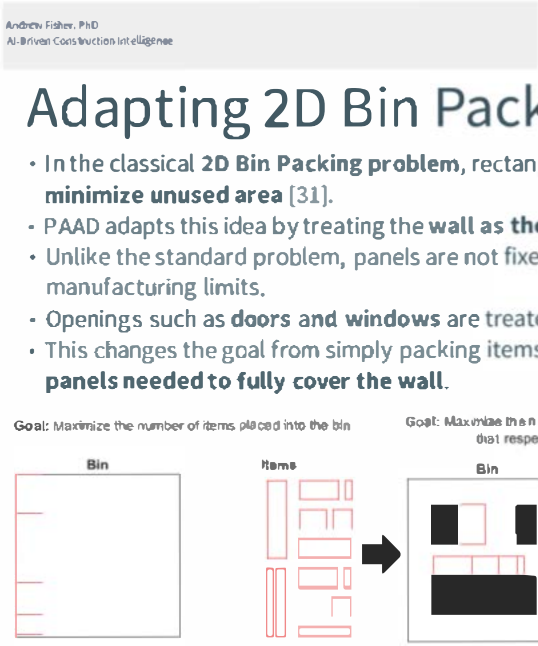
$$
\begin{aligned}
\underset{P, W, O, V, M, C}{\text{minimize}} \quad & fitness(P, W, V, O) \\
\text{subject to} \quad & \\
P = \{p \mid p \in P, evolve\_panels(p, M, C)\}, \quad & \textit{Population evolution}, \\
c_{width}(P) = true, \quad & \textit{Panel width constraint}, \\
c_{height}(P) = true, \quad & \textit{Panel height constraint}
\end{aligned}
$$

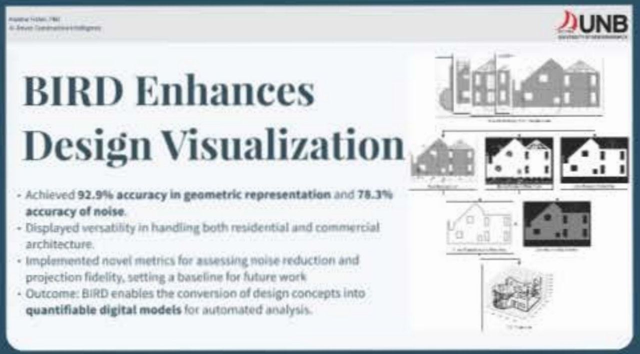
where P is a set of candidate solutions, W is the wall definition, O is the openings definition, V is the void areas definition, M is a mutation probability, and C is a crossover probability.

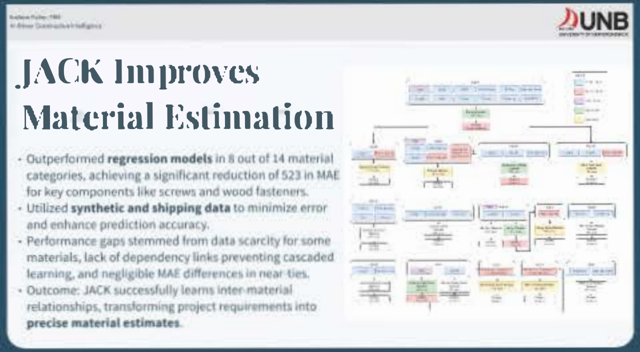
$$
\begin{aligned}
\text{fitness} \;=\; & (\text{wall.width} \cdot \text{wall.height}) = \textit{wall area} \\
& - \sum_{i} (\text{openings}[i].\text{width} \cdot \text{openings}[i].\text{height}) = \textit{openings area} \\
& - \sum_{j} (\text{voids}[j].\text{width} \cdot \text{voids}[j].\text{height}) = \textit{space above angled features' area} \\
& - \sum_{k=0}^{\text{len(panels)}} \begin{cases} \text{panels}[k].\text{width} \cdot \text{panels}[k].\text{height}, & \text{if panels}[k].\text{packed} \\ 0, & \text{otherwise} \end{cases} \\
& + \text{len}\Big(\{\, \text{panel} \mid \text{panel} \in \text{panels}, \text{panel.packed} = \text{true} \,\}\Big)
\end{aligned}
$$

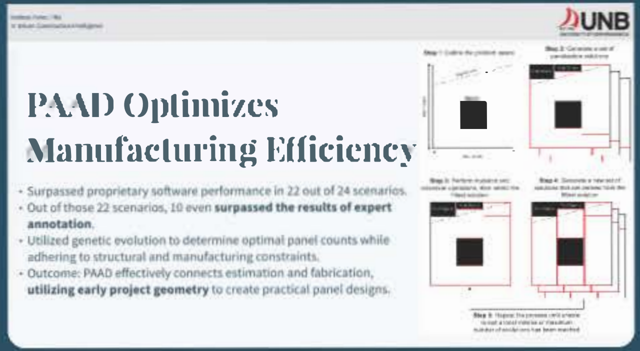

Step 1: Define the problem space
Angled Line
Opening
Wall Height
Wall Width
Step 2: Generate a set of panelization solutions
Void Space
Void Space
Step 3: Perform mutation and crossover operations, then select the fittest solution
Void Space
Void Space
Step 4: Generate a new set of solutions that are derived from the fittest solution
Void Space
Void Space
Step 5: Repeat the process until unable to exit a local minima or maximum number of evolutions has been reached

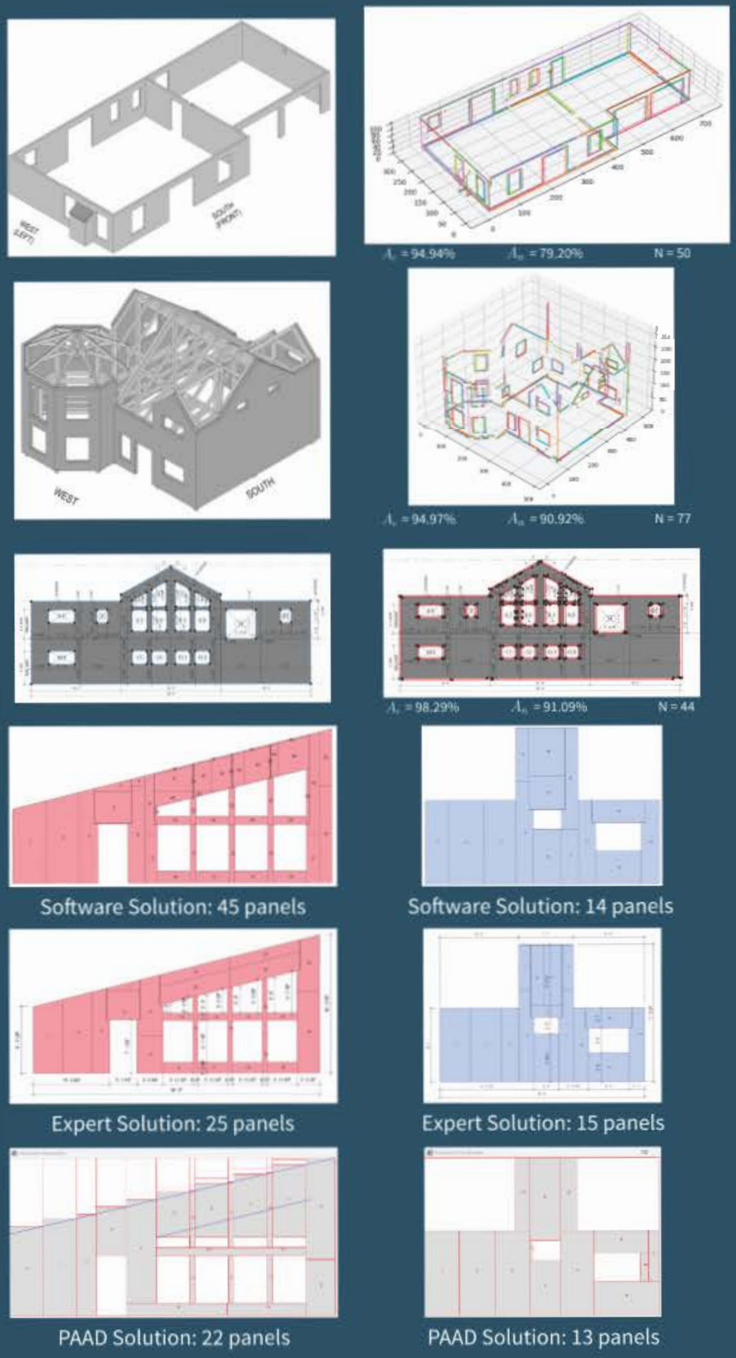

PLOS ONE
RESEARCH ARTICLE
PAAD: Pan
designs
Check for updates


Andrew Fisher, PhD
AI-Driven Construction Intelligence
Adapting 2D Bin Pack
In the classical 2D Bin Packing problem, rectan
minimize unused area [31].
PAAD adapts this idea by treating the wall as the
Unlike the standard problem, panels are not fixe
manufacturing limits.
Openings such as doors and windows are treate
This changes the goal from simply packing items
panels needed to fully cover the wall.
Goal: Maximize the number of items placed into the bin
Bin
Items
Bin







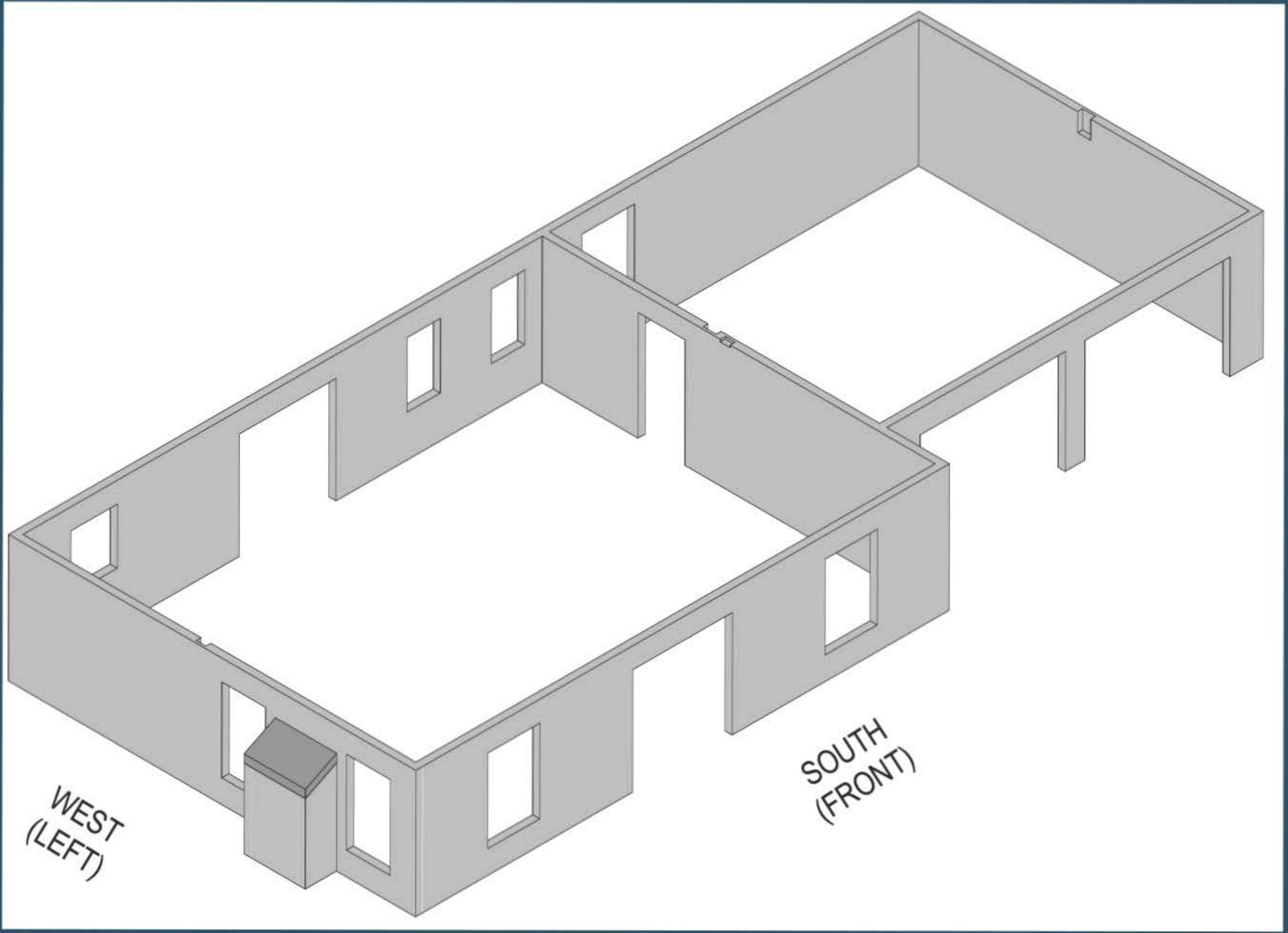


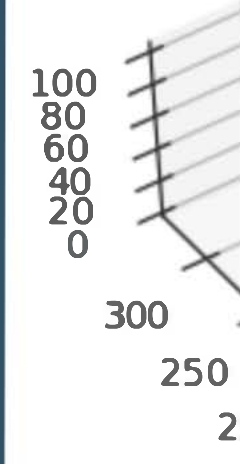
Unified Design-to-Manufacture Flow and Benefits

UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK

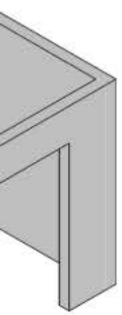
BIRD Enhances Design Visualization

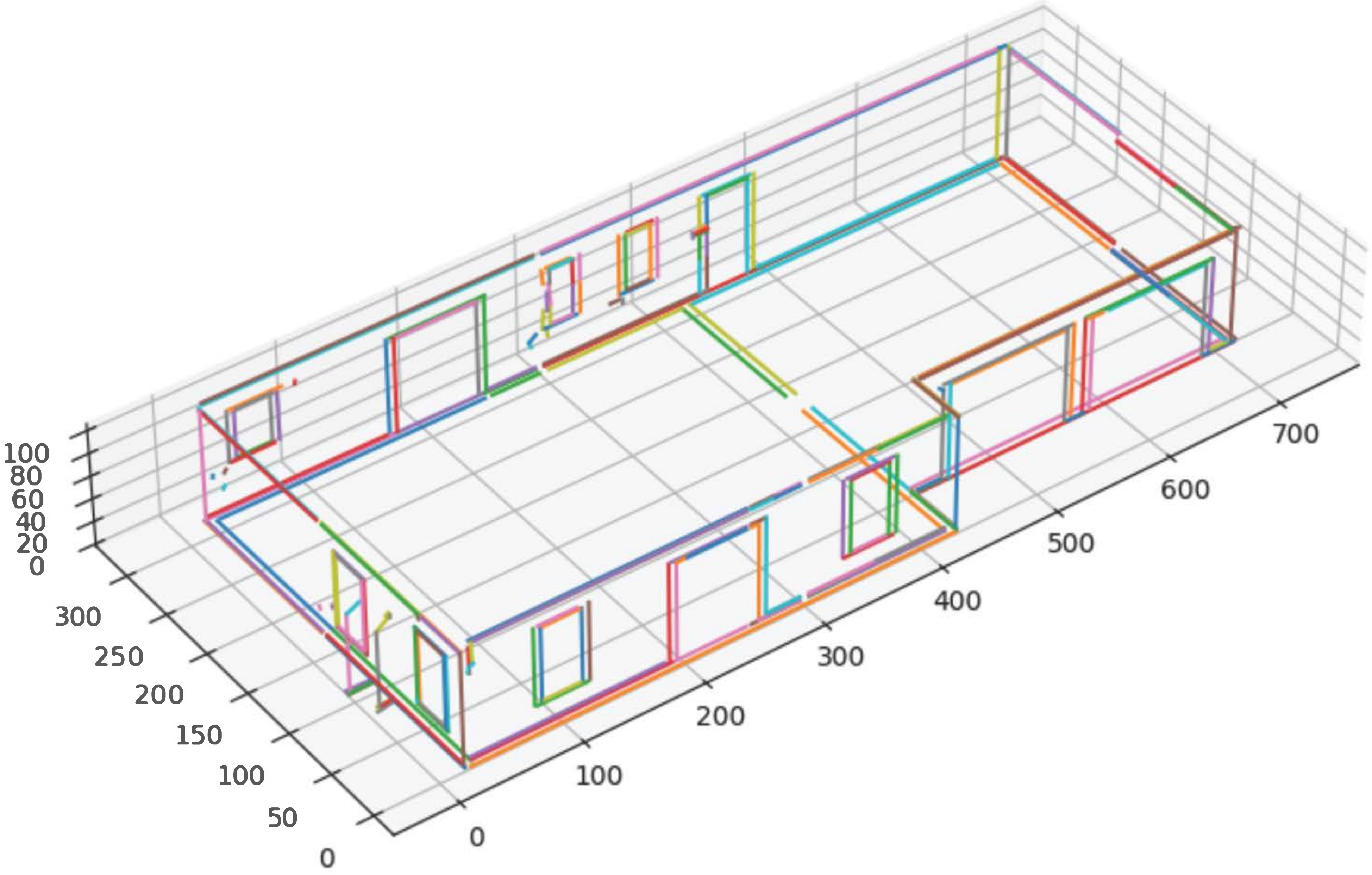
- Achieved **92.9% accuracy in geometric representation** and **78.3% accuracy of noise**.
- Displayed versatility in handling both residential and commercial architecture.
- Implemented novel metrics for assessing noise reduction and projection fidelity, setting a baseline for future work
- Outcome: BIRD enables the conversion of design concepts into **quantifiable digital models** for automated analysis.





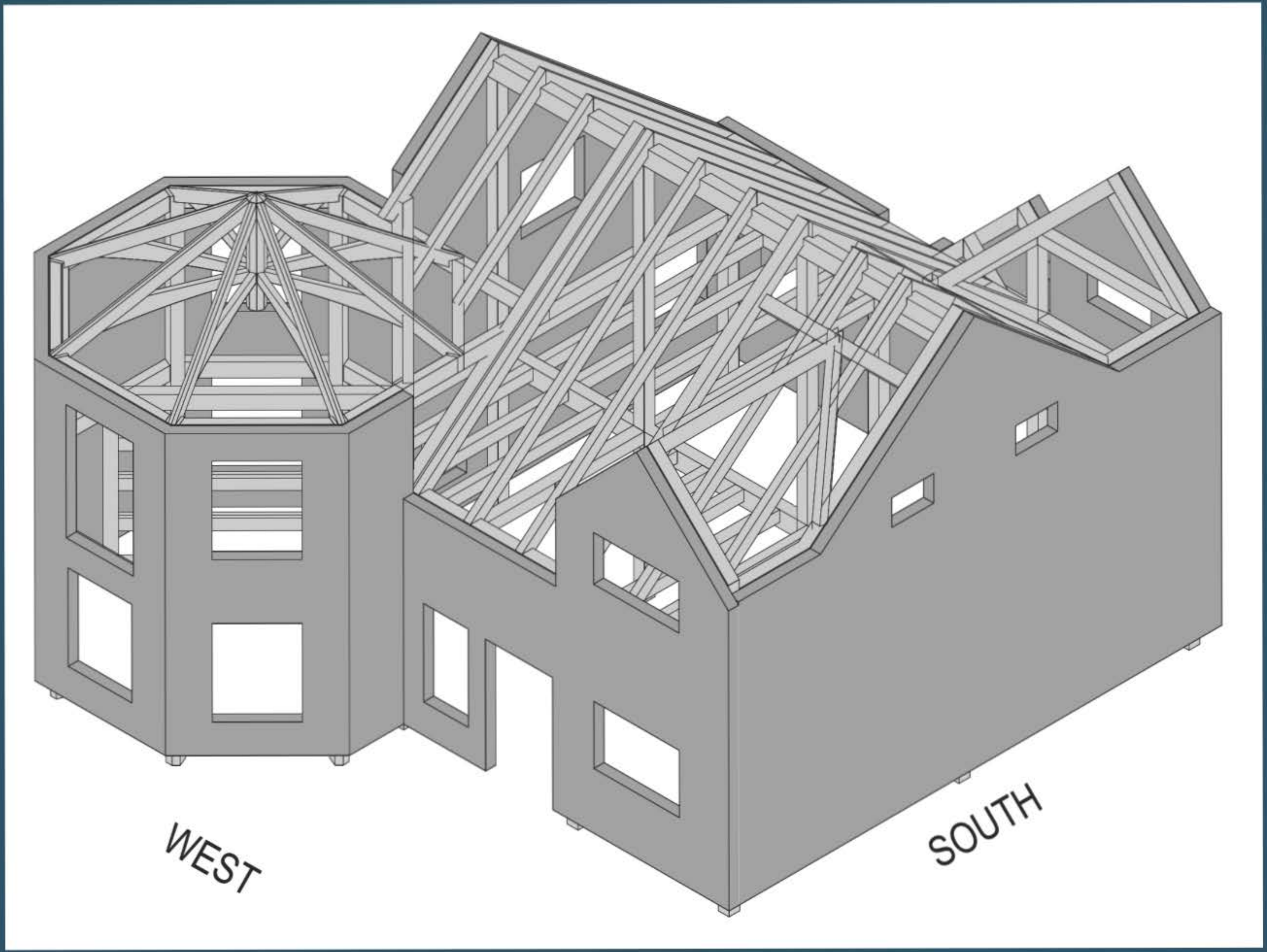
$A_c =$



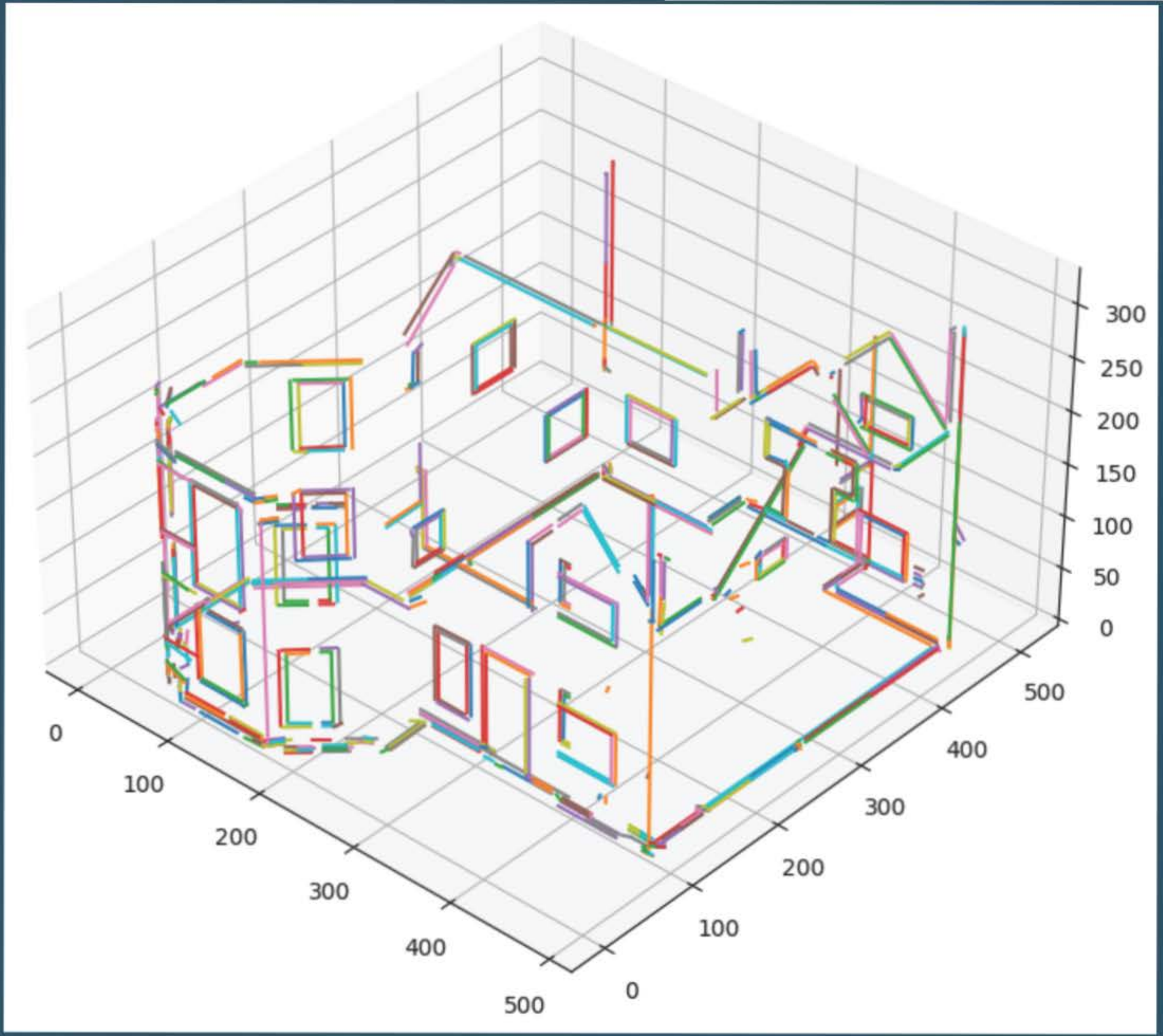


A_c = 94.94% A_n = 79.20% N = 50

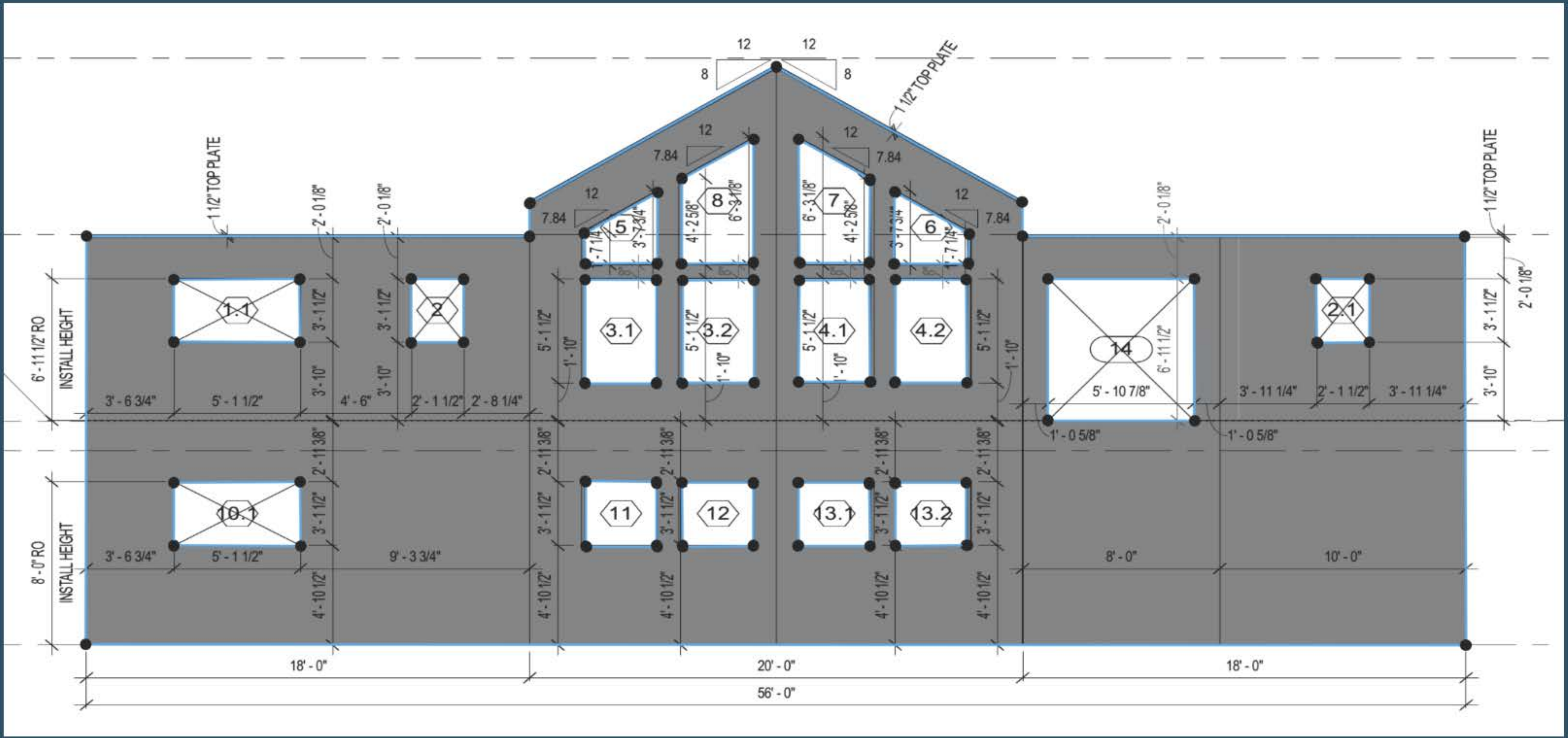
Detection

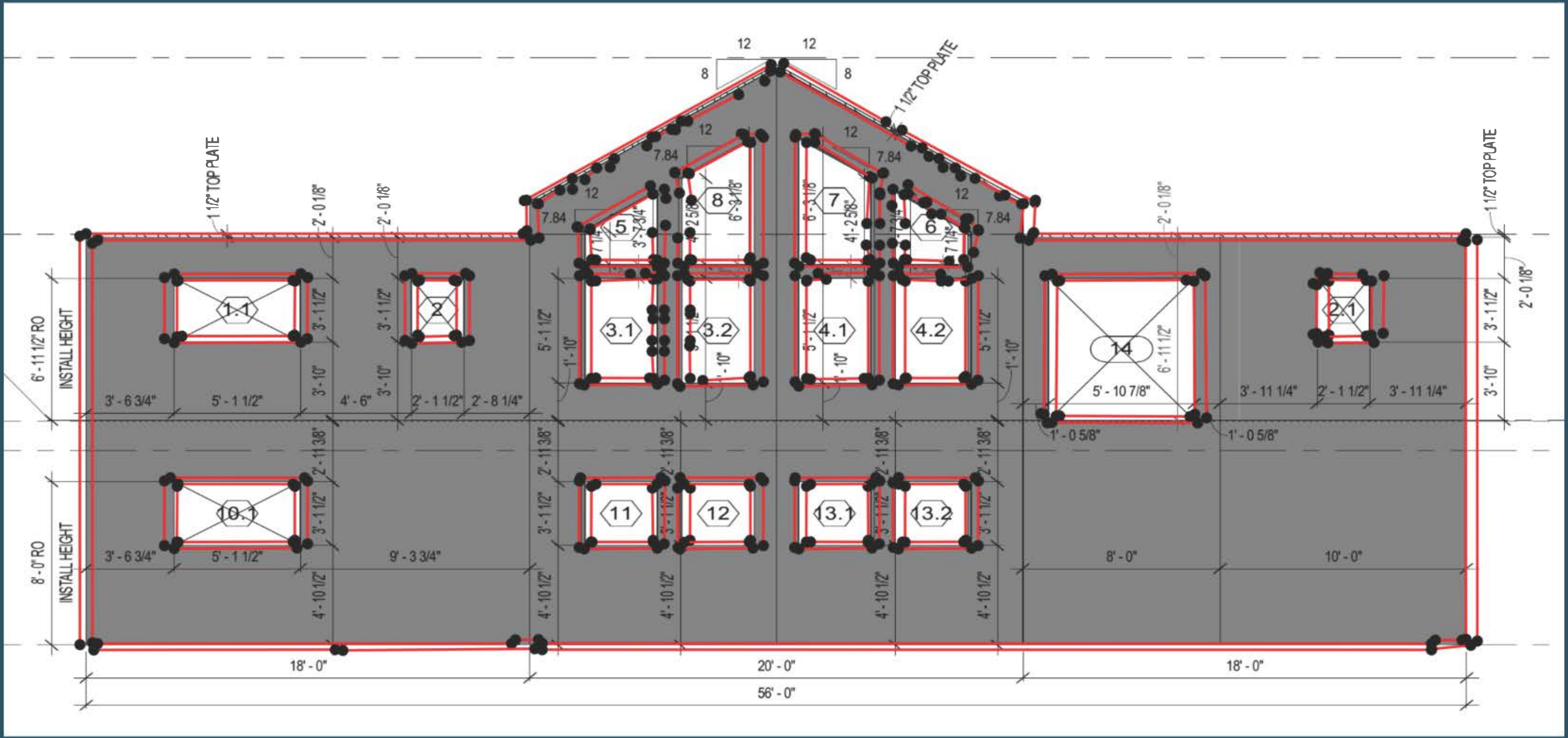


$A_c =$

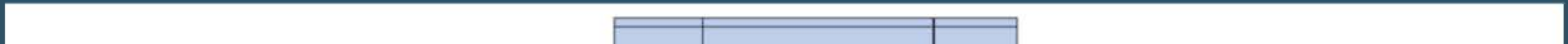


A_c = 94.97% A_n = 90.92% N = 77


1 1/2" TOP PLATE
1 1/2" TOP PLATE
1 1/2" TOP PLATE
6' - 11 1/2" RO
INSTALL HEIGHT
8' - 0" RO
INSTALL HEIGHT
1.1
2
3.1
3.2
4.1
4.2
5
6
7
8
14
2.1
10.1
11
12
13.1
13.2
18' - 0"
20' - 0"
18' - 0"
56' - 0"
3' - 6 3/4"
5' - 1 1/2"
4' - 6"
2' - 1 1/2"
2' - 8 1/4"
9' - 3 3/4"
8' - 0"
10' - 0"
5' - 10 7/8"
1' - 0 5/8"
3' - 11 1/4"



A_c = 98.29% A_n = 91.09% N = 44

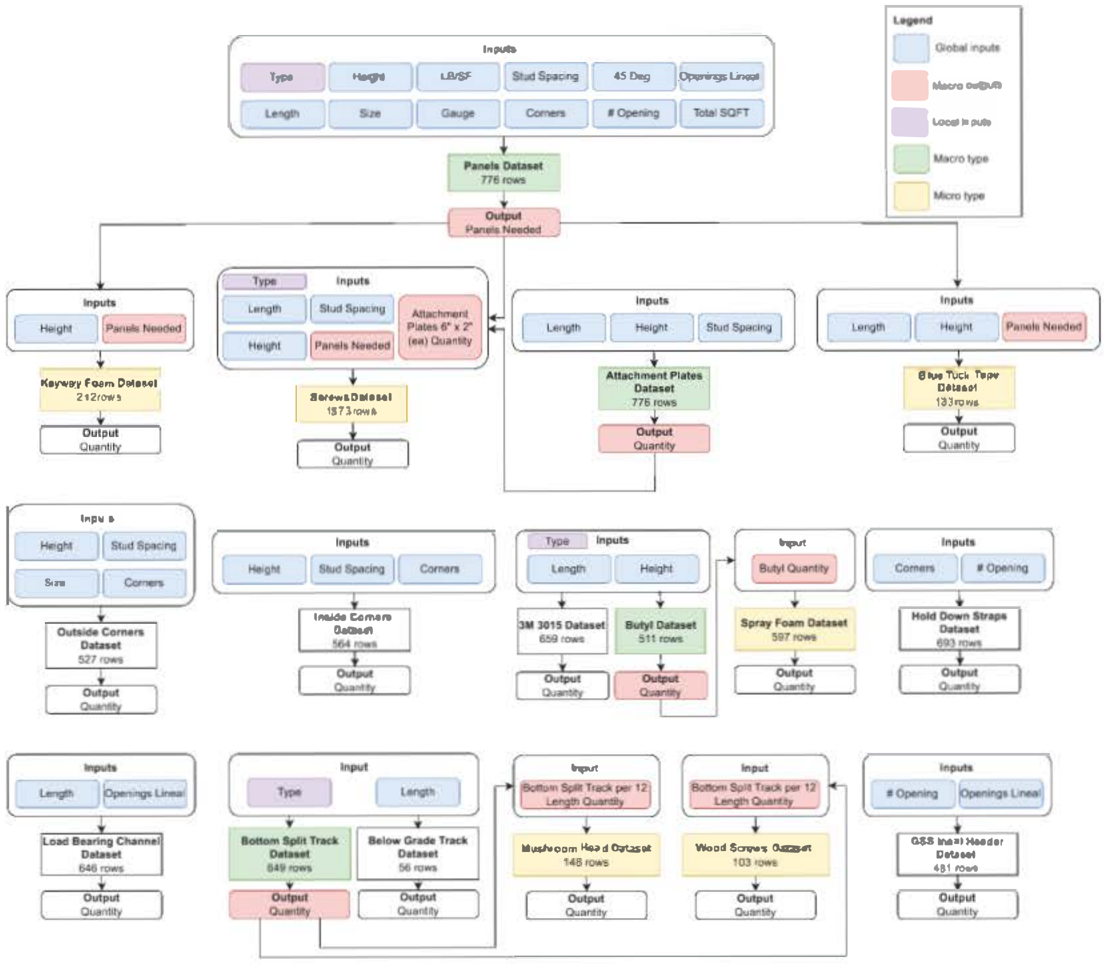


Andrew Fisher, PhD
AI-Driven Construction Intelligence

EST. 1785 UNB
UNIVERSITY OF NEW BRUNSWICK

JACK Improves Material Estimation

- Outperformed **regression models** in 8 out of 14 material categories, achieving a significant reduction of 523 in MAE for key components like screws and wood fasteners.
- Utilized **synthetic and shipping data** to minimize error and enhance prediction accuracy.
- Performance gaps stemmed from data scarcity for some materials, lack of dependency links preventing cascaded learning, and negligible MAE differences in near-ties.
- Outcome: JACK successfully learns inter-material relationships, transforming project requirements into **precise material estimates**.

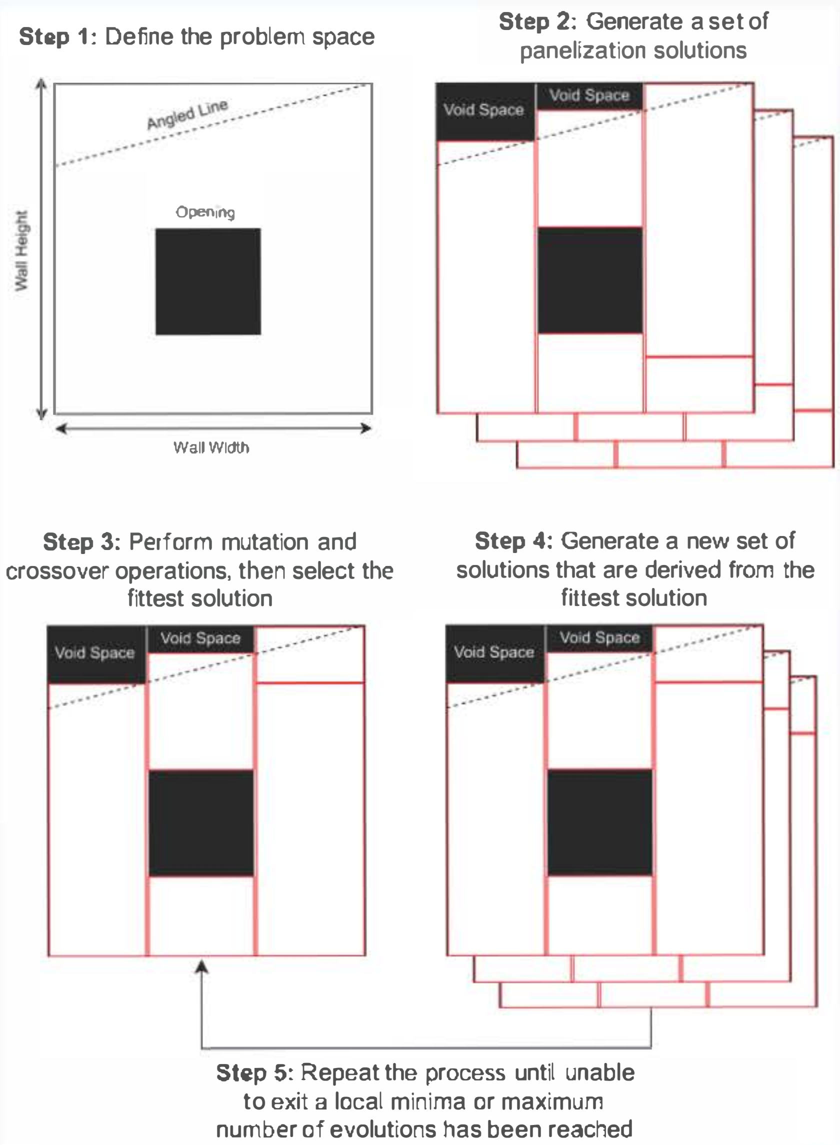


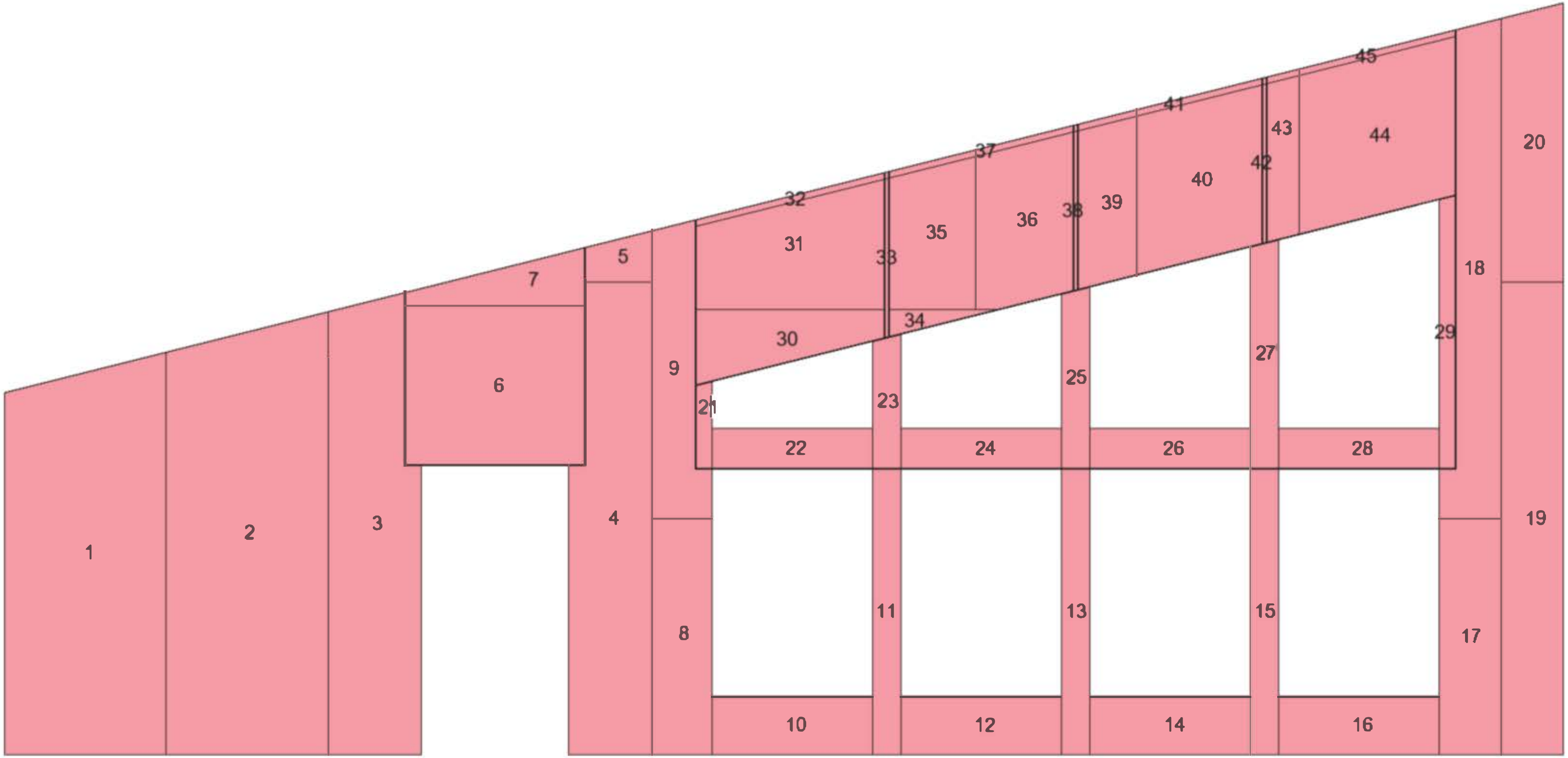
Andrew Fisher, PhD
AI-Driven Construction Intelligence

UNB EST. 1785 UNIVERSITY OF NEW BRUNSWICK

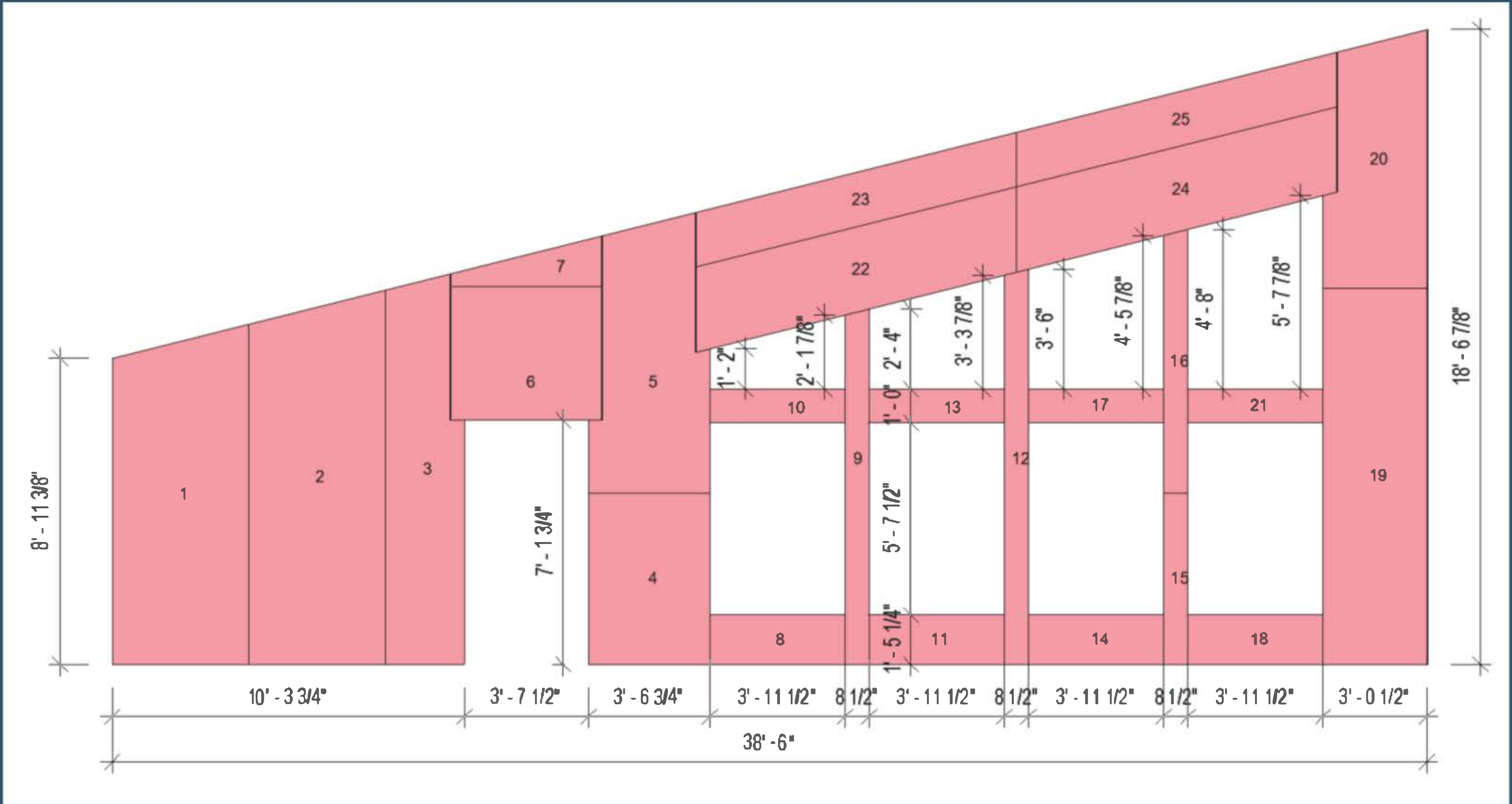
PAAD Optimizes Manufacturing Efficiency

- Surpassed proprietary software performance in 22 out of 24 scenarios.
- Out of those 22 scenarios, 10 even **surpassed the results of expert annotation**.
- Utilized genetic evolution to determine optimal panel counts while adhering to structural and manufacturing constraints.
- Outcome: PAAD effectively connects estimation and fabrication, **utilizing early project geometry** to create practical panel designs.

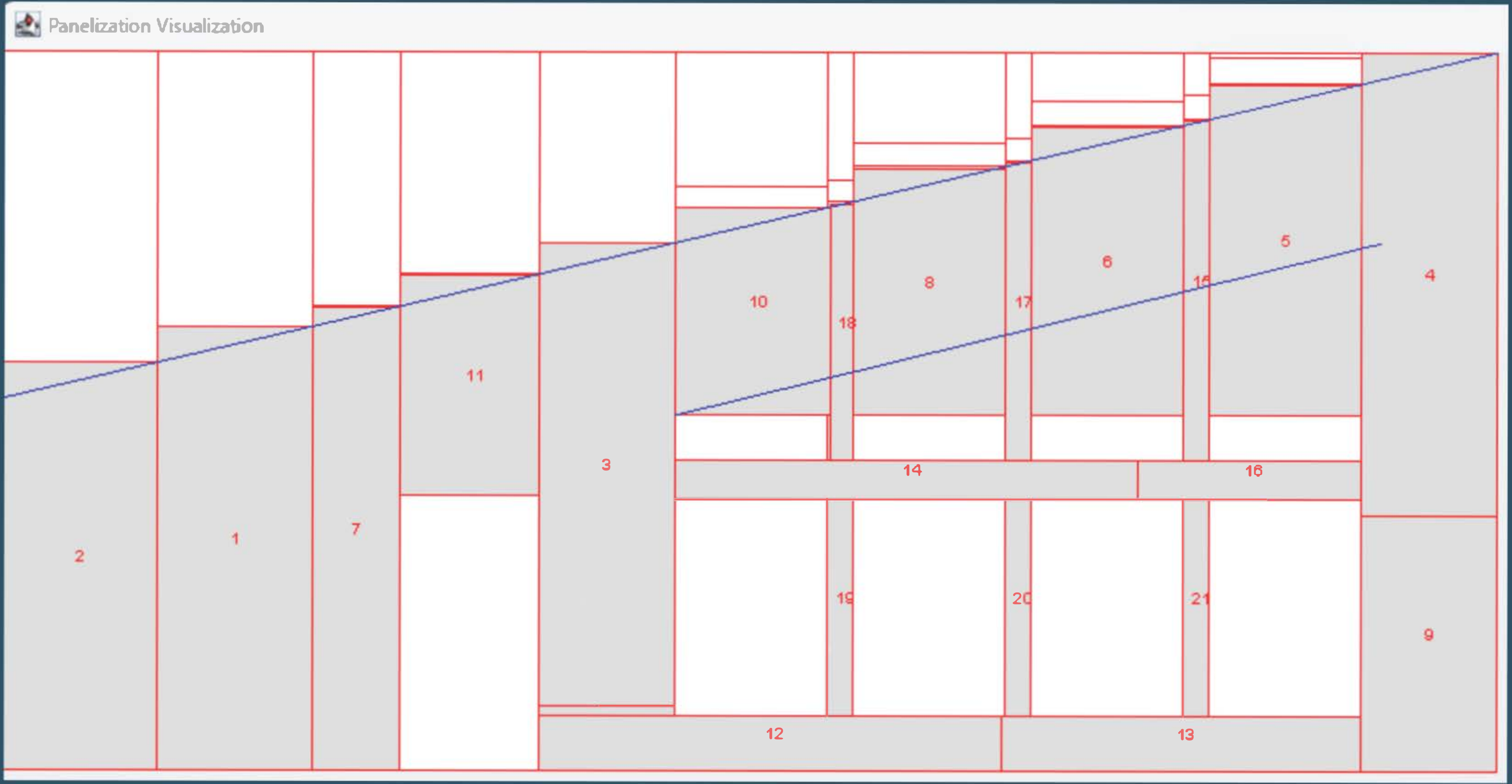




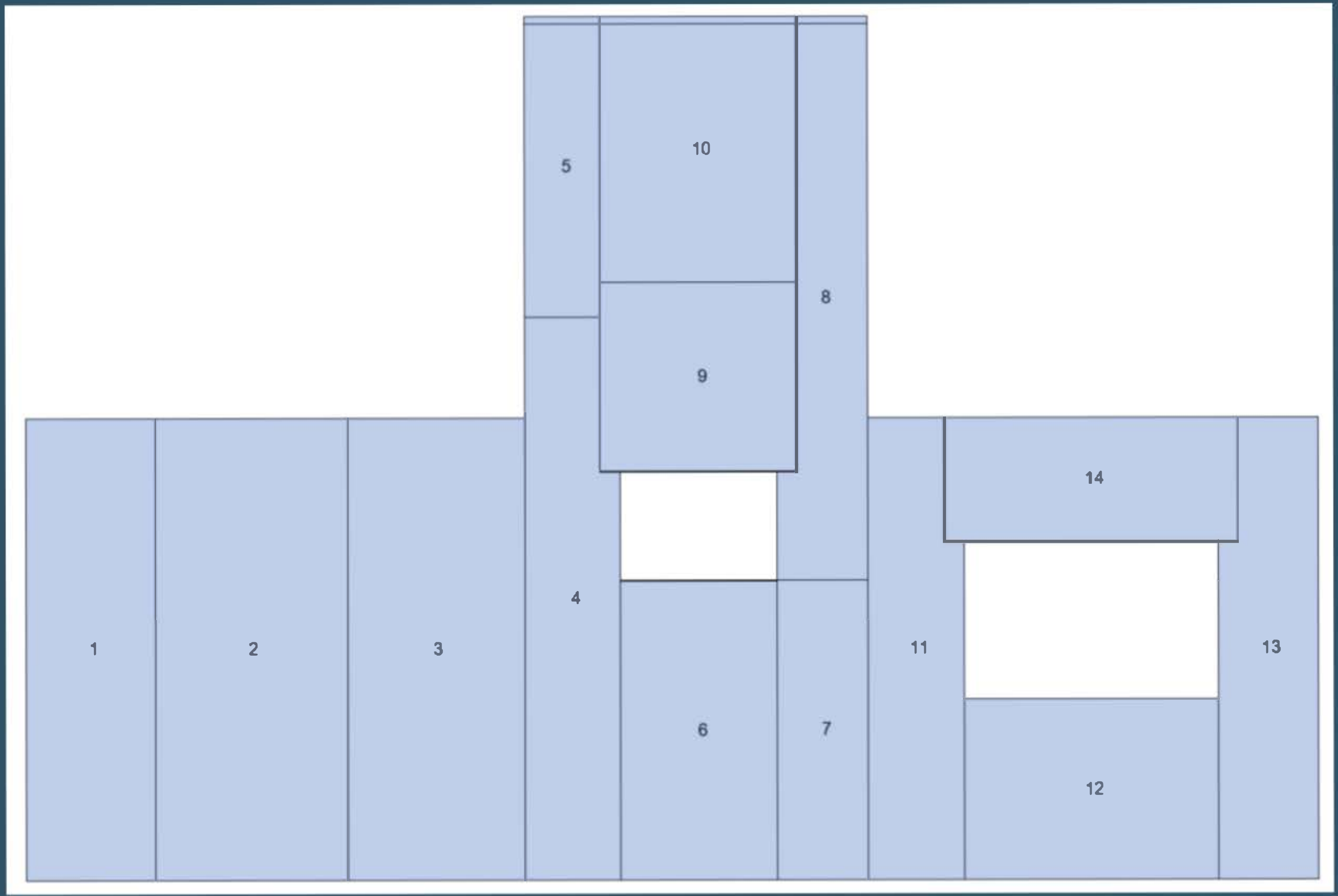
Software Solution: 45 panels



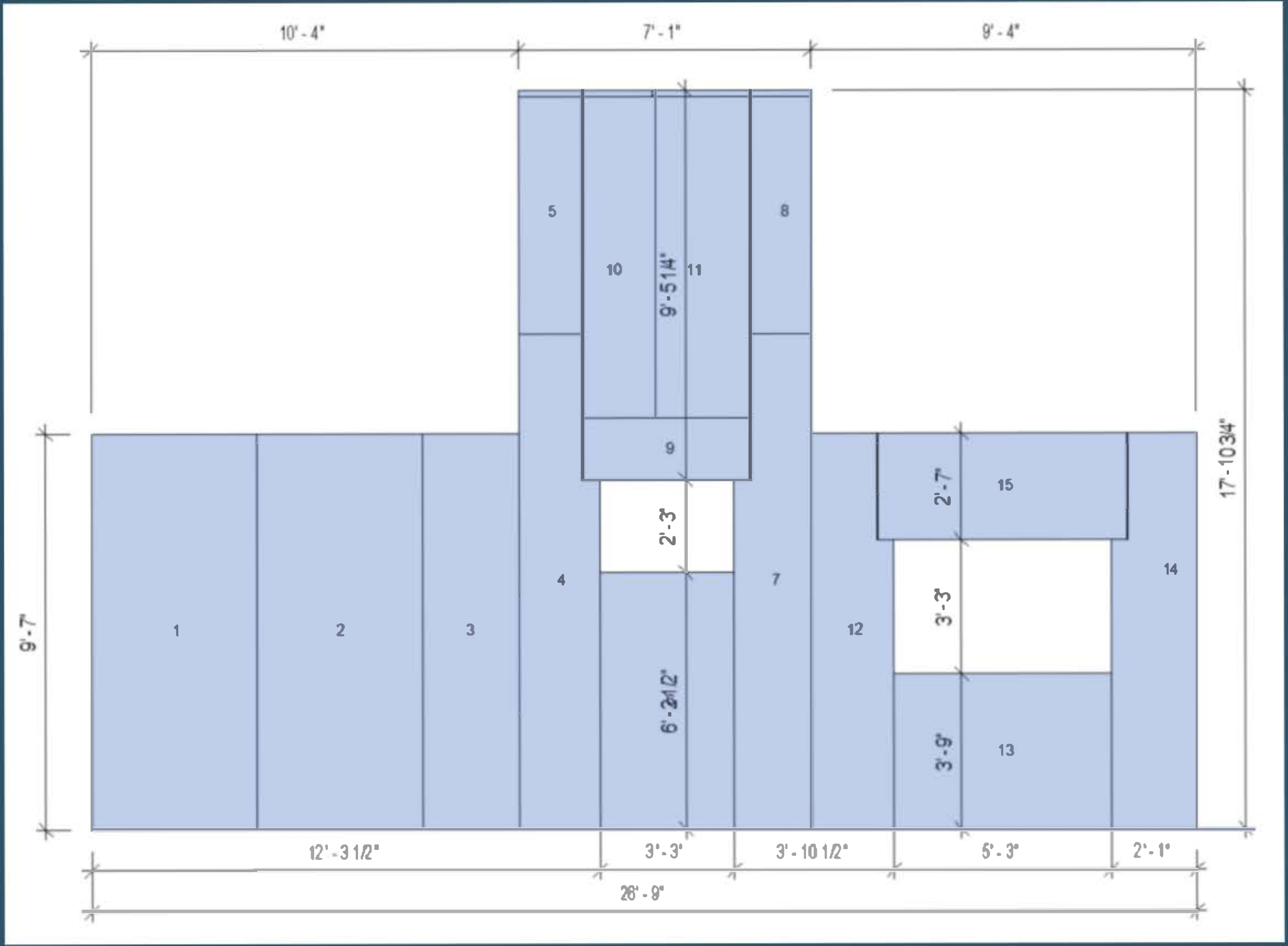
Expert Solution: 25 panels



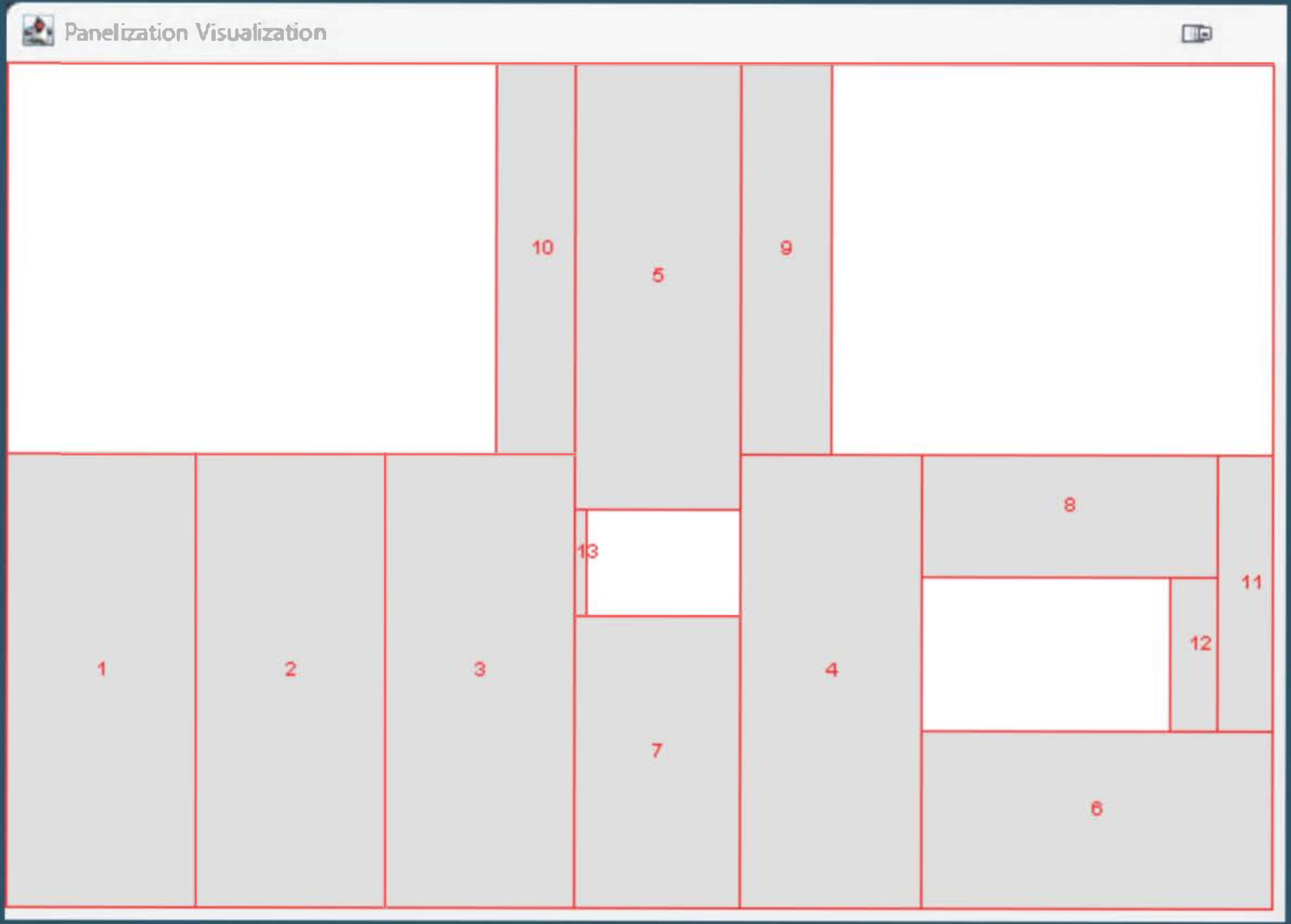
PAAD Solution: 22 panels



Software Solution: 14 panels



Expert Solution: 15 panels



PAAD Solution: 13 panels

Future Directions in AI-Driven Construction Intelligence

UNIVERSITY OF NEW BRUNSWICK

Future Directions in AI-Driven Construction Intelligence

- Transitioning from limited public data to larger, diverse datasets will **enhance the scalability** of frameworks like BIRD, JACK, and PAAD.
- Integrating BIRD, JACK, and PAAD into **an end-to-end automated workflow** will streamline the design-to-manufacture process and improve efficiency.
- Future advancements should focus on making AI systems **transparent and interpretable** to gain industry trust and facilitate collaboration.
- Emphasizing synthetic data generation and benchmarking against established standards will **foster reproducibility and sustainability** in AI applications.
- Creating a collaborative AI infrastructure through integration with BIM systems will pave the way for innovative construction practices.

References

1. Ivson, Paulo, et al. "A systematic review of visualization in building information modeling." *IEEE Transactions on Visualization and Computer Graphics* 26.10 (2019): 3109-3127.
2. Holm, Len, and John E. Schaufelberger. *Construction cost estimating*. Routledge, 2021.
3. Jalaei, Farzad, Milad Zoghi, and Afshin Khoshand. "Life cycle environmental impact assessment to manage and optimize construction waste using Building Information Modeling (BIM)." *International Journal of Construction Management* 21.8 (2021): 784-801.
4. Vidanapathirana, Madhawa, et al. "Plan2scene: Converting floorplans to 3d scenes." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
5. Liu, Hao, et al. "A novel Data-Driven framework based on BIM and knowledge graph for automatic model auditing and Quantity Take-off." *Advanced Engineering Informatics* 54 (2022): 101757.
6. Slowik, Adam, and Halina Kwasnicka. "Evolutionary algorithms and their applications to engineering problems." *Neural Computing and Applications* 32.16 (2020): 12363-12379.
7. Luo, Lan, Limao Zhang, and Qinghua He. "Linking project complexity to project success: A hybrid SEM–FCM method." *Engineering, Construction and Architectural Management* 27.9 (2020): 2591-2614.
8. Cuellar Lobo, Jose Daniel, et al. "Building Information Modelling-(BIM-) Based Generative Design for Drywall Installation Planning in Prefabricated Construction." *Advances in Civil Engineering* 2021.1 (2021): 6638236.
9. Fisher, Andrew, et al. "Building image reconstruction and dimensioning of the envelope from two-dimensional perspective drawings." *Engineering Applications of Artificial Intelligence* 139 (2025): 109657.
10. Fisher, Andrew, et al. "Jointly Trained Automation of Explainable Construction Material Knowledge." Expert Systems 43.3 (2026): e70209.
11. Fisher, Andrew, et al. "PAAD: Panelization algorithm for architectural designs." *PLOS One* 19.6 (2024): e0303646.
12. Ma, Li, and Hongwei Fu. "Exploring the influence of project complexity on the mega construction project success: a qualitative comparative analysis (QCA) method." *Engineering, Construction and Architectural Management* 27.9 (2020): 2429-2449.
13. Bozic, Aljaz, et al. "Transformerfusion: Monocular rgb scene reconstruction using transformers." *Advances in Neural Information Processing Systems* 34 (2021): 1403-1414.
14. Patil, Vaishakh, et al. "P3depth: Monocular depth estimation with a piecewise planarity prior." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
15. Shue, J. Ryan, et al. "3d neural field generation using triplane diffusion." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
16. Xie, Saining, and Zhuowen Tu. "Holistically-nested edge detection." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.
17. Goel, Nidhi, Samarjeet Kaur, and Ruchika Bala. "Dual branch convolutional neural network for copy move forgery detection." IET Image Processing 15.3 (2021): 656-665.
18. Liao, Minghui, Baoguang Shi, and Xiang Bai. "Textboxes++: A single-shot oriented scene text detector." *IEEE Transactions on Image Processing* 27.8 (2018): 3676-3690.
19. Dumoulin, Vincent, and Francesco Visin. "A guide to convolution arithmetic for deep learning." *arXiv preprint* arXiv:1603.07285 (2016).
20. Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research* 15.1 (2014): 1929-1958.
21. Abanda, Fonbeyin Henry, B. Kamsu-Foguem, and Joseph Handibry Mbatu Tah. "BIM–New rules of measurement ontology for construction cost estimation." *Engineering Science and Technology, an International Journal* 20.2 (2017): 443-459.
22. Tayefeh Hashemi, Sanaz, Omid Mahdi Ebadati, and Harleen Kaur. "Cost estimation and prediction in construction projects: A systematic review on machine learning techniques." *SN Applied Sciences* 2.10 (2020): 1703.
23. Elfahham, Yasser. "Estimation and prediction of construction cost index using neural networks, time series, and regression." *Alexandria Engineering Journal* 58.2 (2019): 499-506.
24. Yoo, Soyoung, and Namwoo Kang. "Explainable artificial intelligence for manufacturing cost estimation and machining feature visualization." *Expert Systems with Applications* 183 (2021): 115430.
25. Shen, Yun, et al. "Joint training and resource allocation optimization for federated learning in UAV swarm." *IEEE Internet of Things Journal* 10.3 (2022): 2272-2284.
26. Marquez, Enrique S., Jonathon S. Hare, and Mahesan Niranjan. "Deep cascade learning." *IEEE Transactions on Neural Networks and Learning Systems* 29.11 (2018): 5475-5485.
27. Zhang, Nan, et al. "BIM-based automated design of drainage systems for panelized residential buildings." *International Journal of Construction Management* 23.15 (2023): 2683-2698.
28. Dayama, Niraj Ramesh, et al. "Grids: Interactive layout design with integer programming." *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020.
29. Becker, Henrique, et al. "Comparative analysis of mathematical formulations for the two-dimensional guillotine cutting problem." *International Transactions in Operational Research* 31.5 (2024): 3010-3035.
30. Ahn, SangJun, SangUk Han, and Mohamed Al-Hussein. "2D drawing visualization framework for applying projection-based augmented reality in a panelized construction manufacturing facility: Proof of concept." *Journal of Computing in Civil Engineering* 33.5 (2019): 04019032.
31. Polyakovskiy, Sergey, and Rym M'Hallah. "Just-in-time two-dimensional bin packing." *Omega* 102 (2021): 102311.
32. Bossek, Jakob, et al. "Evolving diverse TSP instances by means of novel and creative mutation operators." *Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*. 2019.

UNB
EST. 1785
UNIVERSITY OF NEW BRUNSWICK

Thank you for listening! Any questions?

Feel free to contact me at:

andrew.fisher@unb.ca

Engineering Applications of Artificial Intelligence 139 (2025) 109657
Contents lists available at ScienceDirect
Engineering Applications of Artificial Intelligence
journal homepage: www.elsevier.com/locate/engappai

Building image reconstruction and dimensioning of the envelope from two-dimensional perspective drawings

Andrew Fisher, Lucas Moreira, Muntasir Billah, Pawan Lingras, Vijay Mago

Expert Systems
WILEY
ORIGINAL ARTICLE OPEN ACCESS

Jointly Trained Automation of Explainable Construction Material Knowledge

Andrew Fisher | Lucas Moreira | Muntasir Billah | Pawan Lingras | Vijay Mago

RESEARCH ARTICLE

PAAD: Panelization algorithm for architectural designs

Andrew Fisher, Xing Tan, Muntasir Billah, Pawan Lingras, Jimmy Huang, Vijay Mago

OPEN ACCESS

Abstract

UNB
EST. 1785
UNIVERSITY OF NEW BRUNSWICK

AI-Driven Construction Intelligence

From Design Understanding to Manufacturing Optimization

Andrew Fisher, PhD

April. 30, 2026

OCRC Webinar Series