Time-Aware Packing for Faster 3D Printing Throughput

Blayne Montaque, Stephone Christian, Saurav K. Aryal PhD
AI4PC Lab, Institute for Human Centered Artificial Intelligence

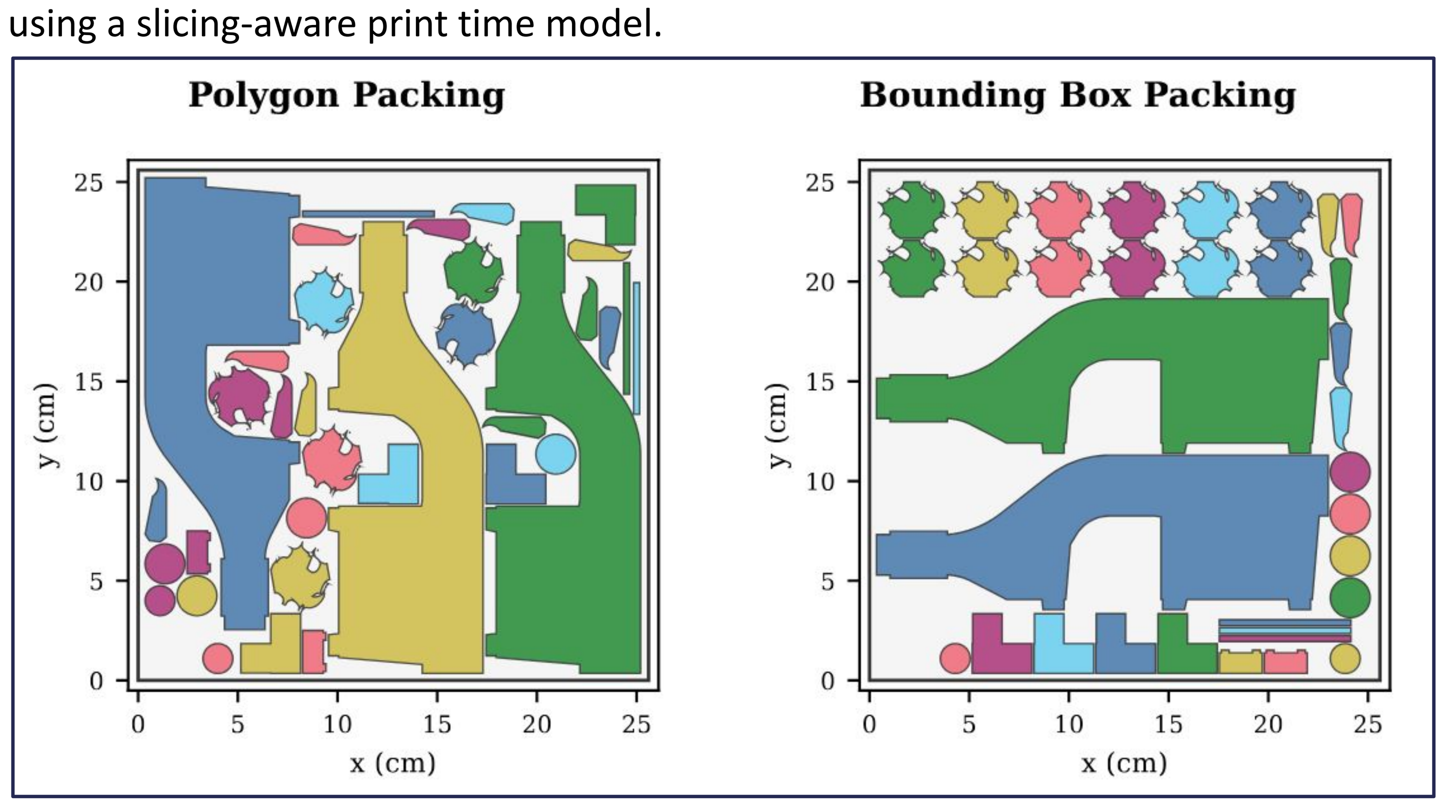
INTRODUCTION

3D printing workflows often require producing multiple parts for a single assembly. A common strategy is to batch parts on one build plate to improve efficiency.

Most packing algorithms optimize for **geometric fit — not actual print time**.

In practice, print time depends on:
- Travel between parts
- Layer-by-layer interactions
- Slicer toolpath behavior

We introduce a **time-aware packing approach** that directly optimizes throughput using a slicing-aware print time model.



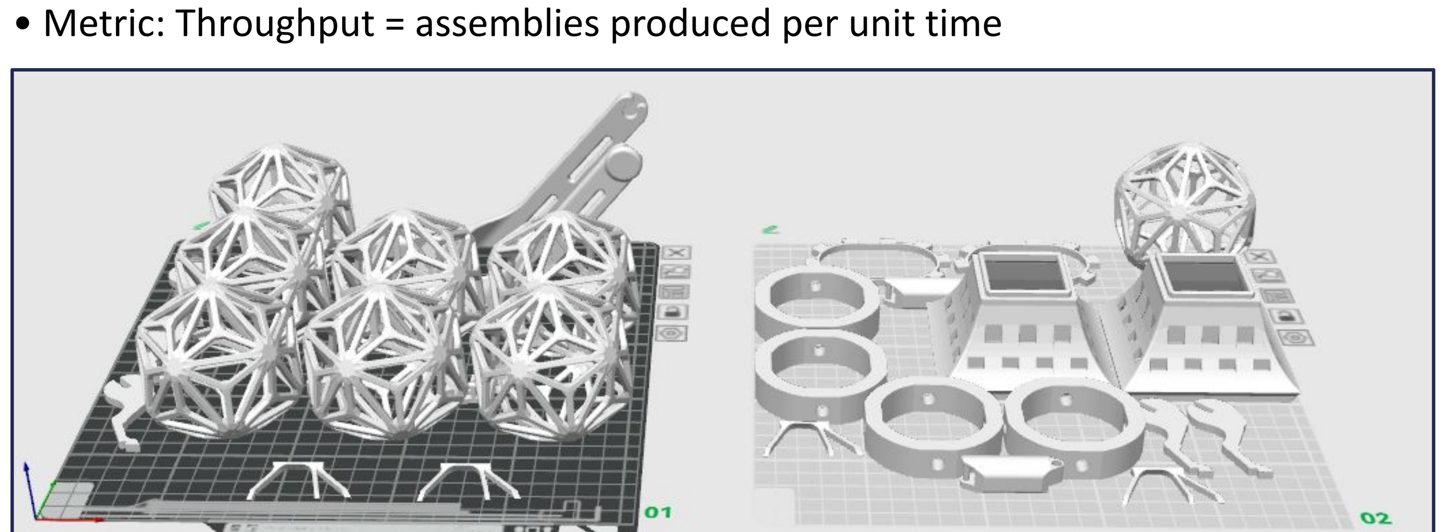
Different layouts → different travel paths → different print times

EXPERIMENTAL SETUP

Dataset:
- 4,187 STL models from Thingi10K
- Filtered for Bambu X1C build volume

Build Generation:
- 3–15 unique parts per build
- 10–150 total parts after replication
- Realistic production-style batches

Evaluation:
- Compared packing algorithms:
  - Bounding Box (fast baseline)
  - Polygon Projection (proposed)
  - Large Neighborhood Search (LNS)
- Baseline: PrusaSlicer auto-arrange
- Metric: Throughput = assemblies produced per unit time



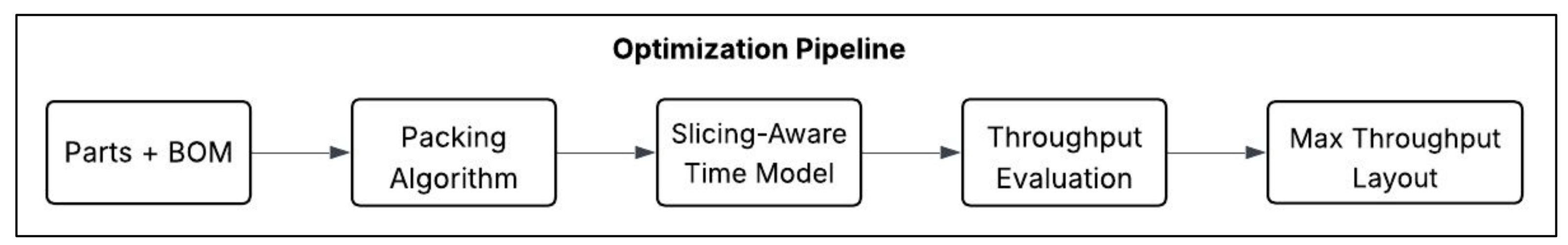
APPROACH

Goal:
Maximize throughput over a fixed time horizon

Key Idea:
Packing decisions affect print time - not just geometric fit

Pipeline:
1. Select part quantities (x)
2. Generate feasible 2D packing
3. Estimate print time (without toolpath generation)
4. Optimize layout for maximum throughput

Output:
Best packing configuration that maximizes completed assemblies per unit time



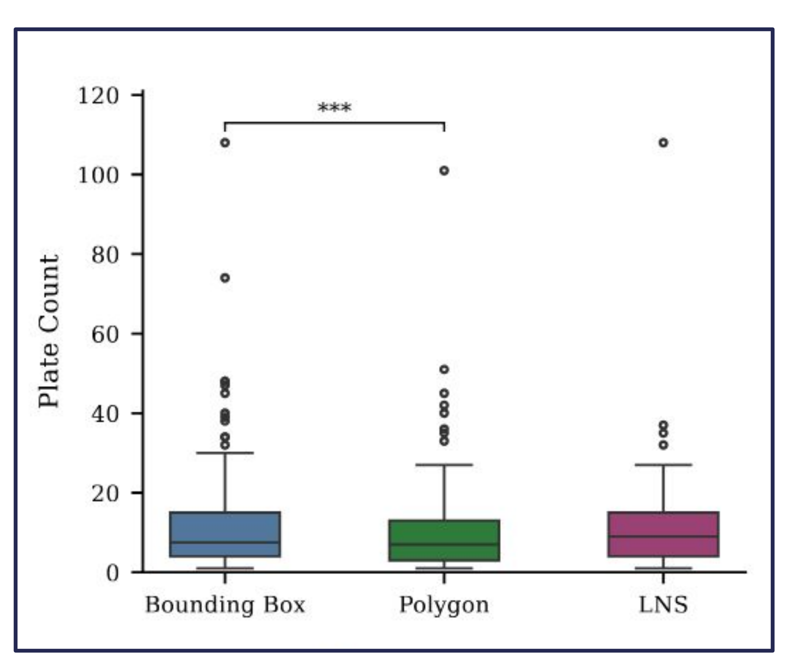
No toolpath generation required

ALGORITHM INSIGHTS

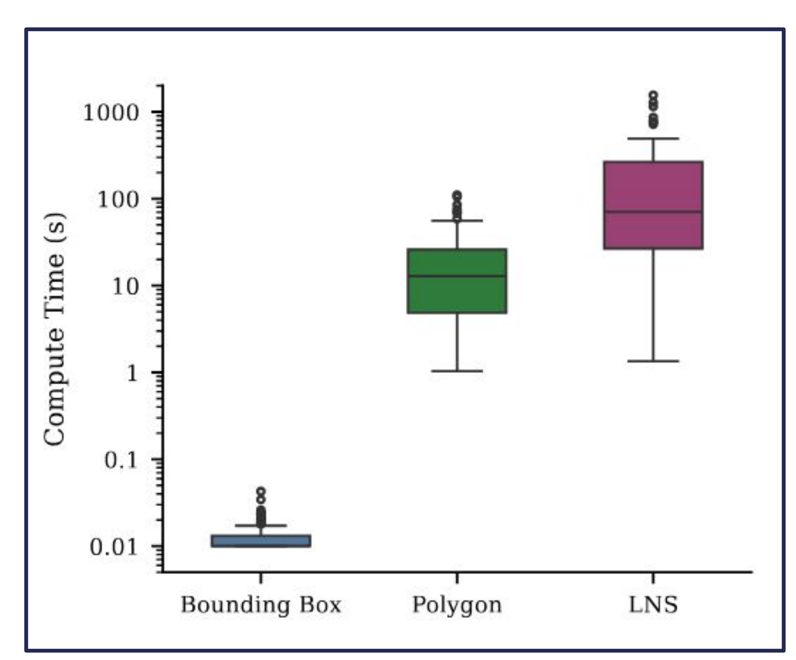
Algorithm Insights:
- Bounding Box: **fastest** (<1 s median), but **uses more plates**
- Polygon Projection: **13.5% fewer plates** than Bounding Box
- LNS: ~71 s median compute, no throughput improvement

Key Finding:
More expensive optimization **did not produce better layouts**.

Why?
Greedy packers already achieve high initial packing density, leaving little room for refinement.



Polygon packing reduces plate count compared to bounding box



LNS is significantly slower with no throughput benefit

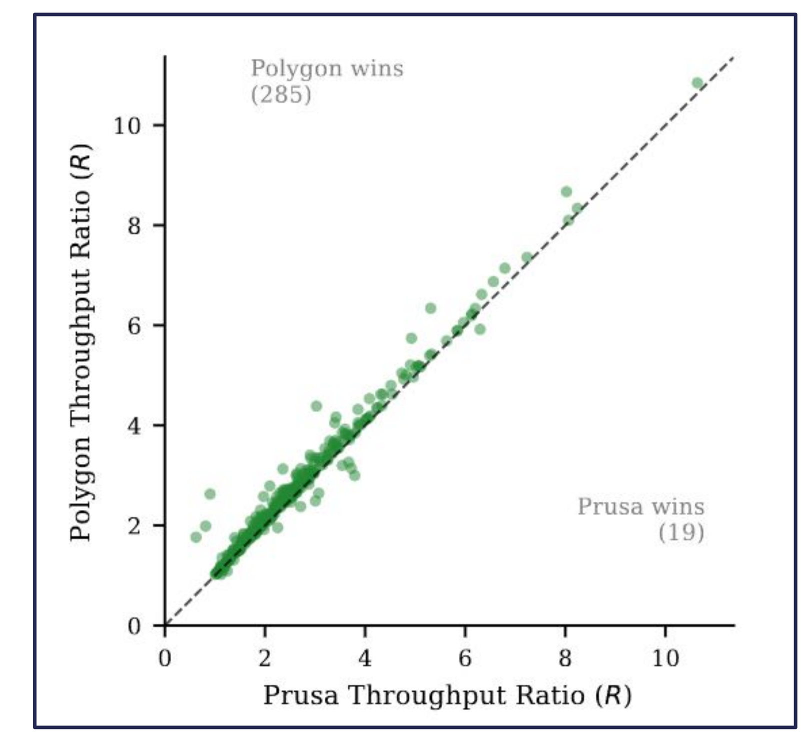
For real-world 3D printing:
- Use fast greedy packing methods
- Avoid expensive metaheuristics
- Optimize for print time — not just geometry

→ Better throughput with minimal compute cost

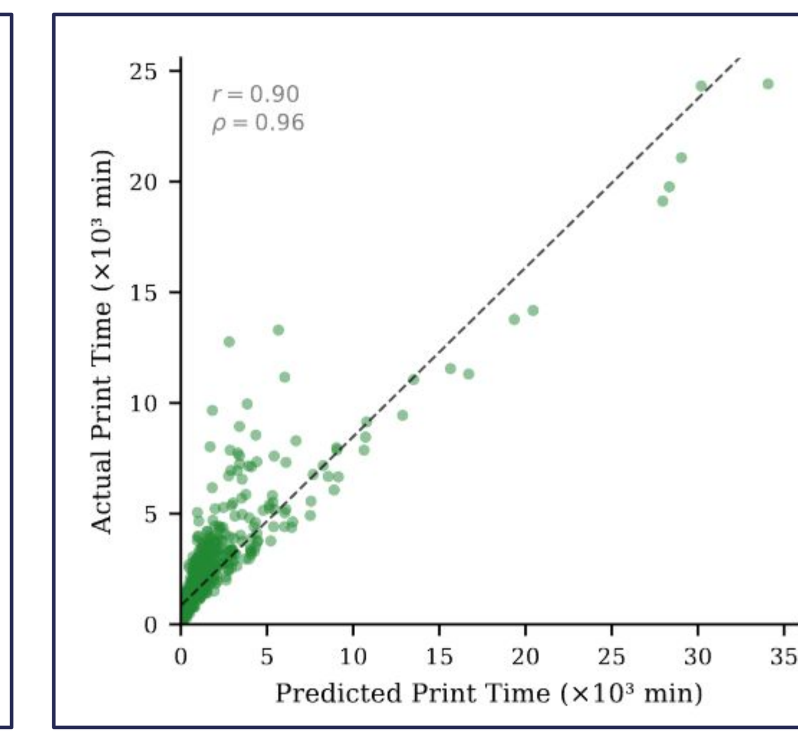
RESULTS

Key Results:
- **+5.7% throughput improvement** (vs Prusa auto-arrange)
- Median savings: **~19.5 min** per build
- Comparable compute time (~3–10 seconds)

Model Validation:
- Strong agreement with slicer estimates
- Pearson r = 0.90
- Spearman ρ = 0.96



Most builds show improved throughput (points above diagonal)



Strong agreement between predicted and actual print times validates the time model

CONCLUSION

Key Points:
- Time-aware packing improves throughput (+5.7%)
- ~19.5 minutes saved per build
- No toolpath generation required
- Greedy methods are sufficient for high-performance packing.
- Enables faster, more efficient 3D printing workflows.

REFERENCES

- Zhou, Q., & Jacobson, A. (2016). Thingi10K: A Dataset of 10,000 3D-Printing Models. arXiv:1605.04797
- PrusaSlicer (Slic3r-based). Open-source 3D printing slicer
- Jylänki, J. (2010). A Thousand Ways to Pack the Bin
- Bennell, J. A., & Oliveira, J. F. (2008). The Geometry of Nesting Problems
- Marler, R. T., & Arora, J. S. (2004). Multi-Objective Optimization Methods
- Shaw, P. (1998). Local Search Methods for Routing Problems

ACKNOWLEDGEMENTS

This project was supported in part by the Office of Naval Research grant N00014-22-1-2714. The work is solely the responsibility of the authors and does not necessarily represent the official view of the Office of Naval Research.