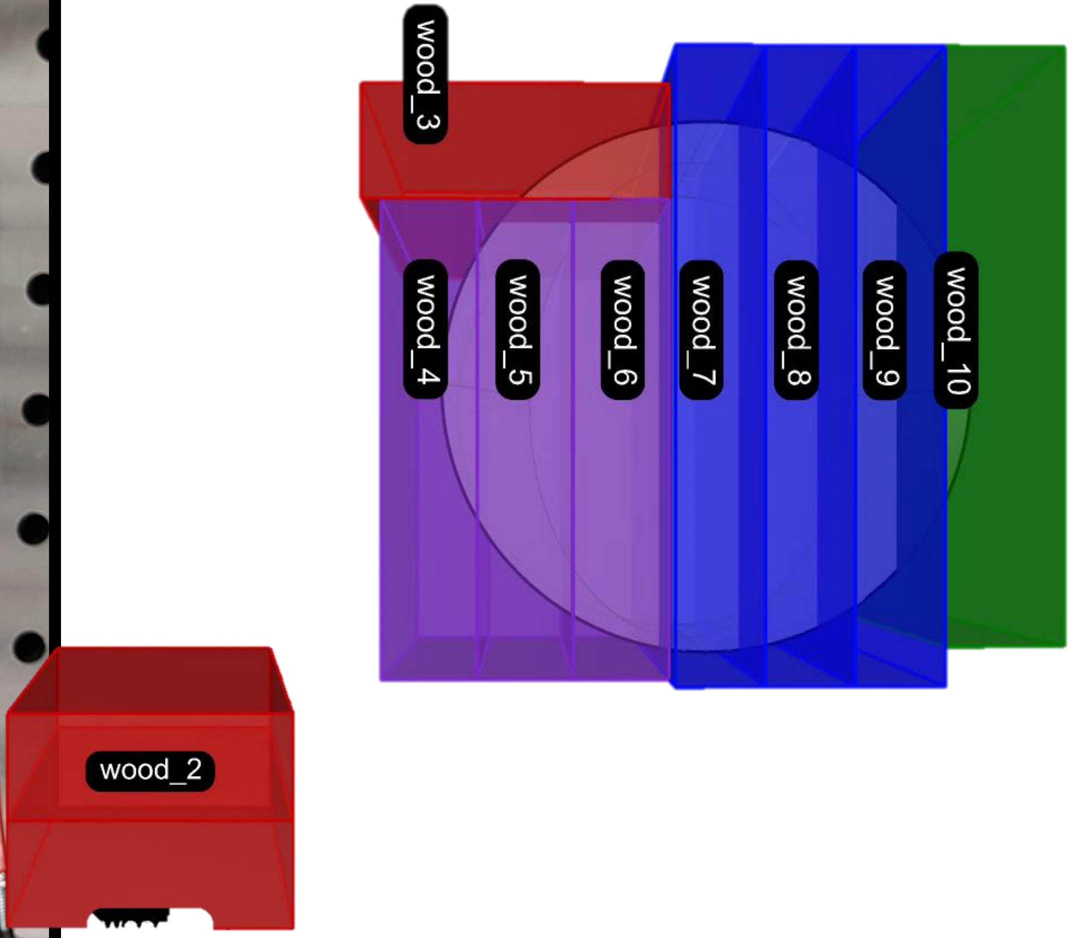
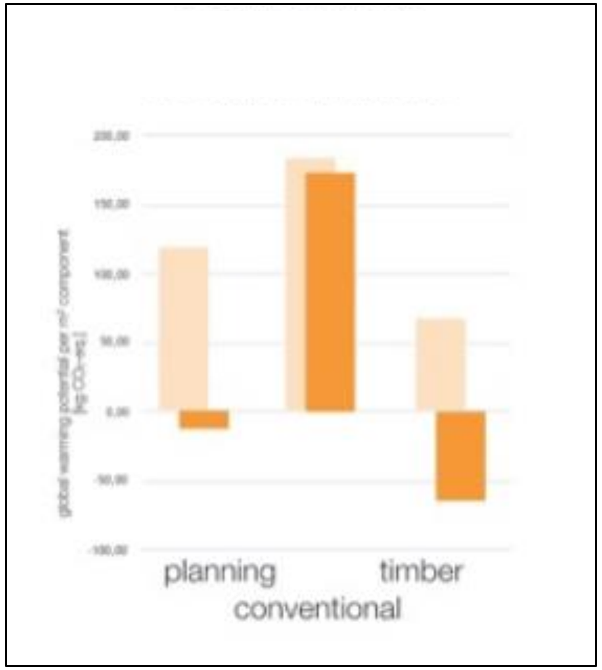
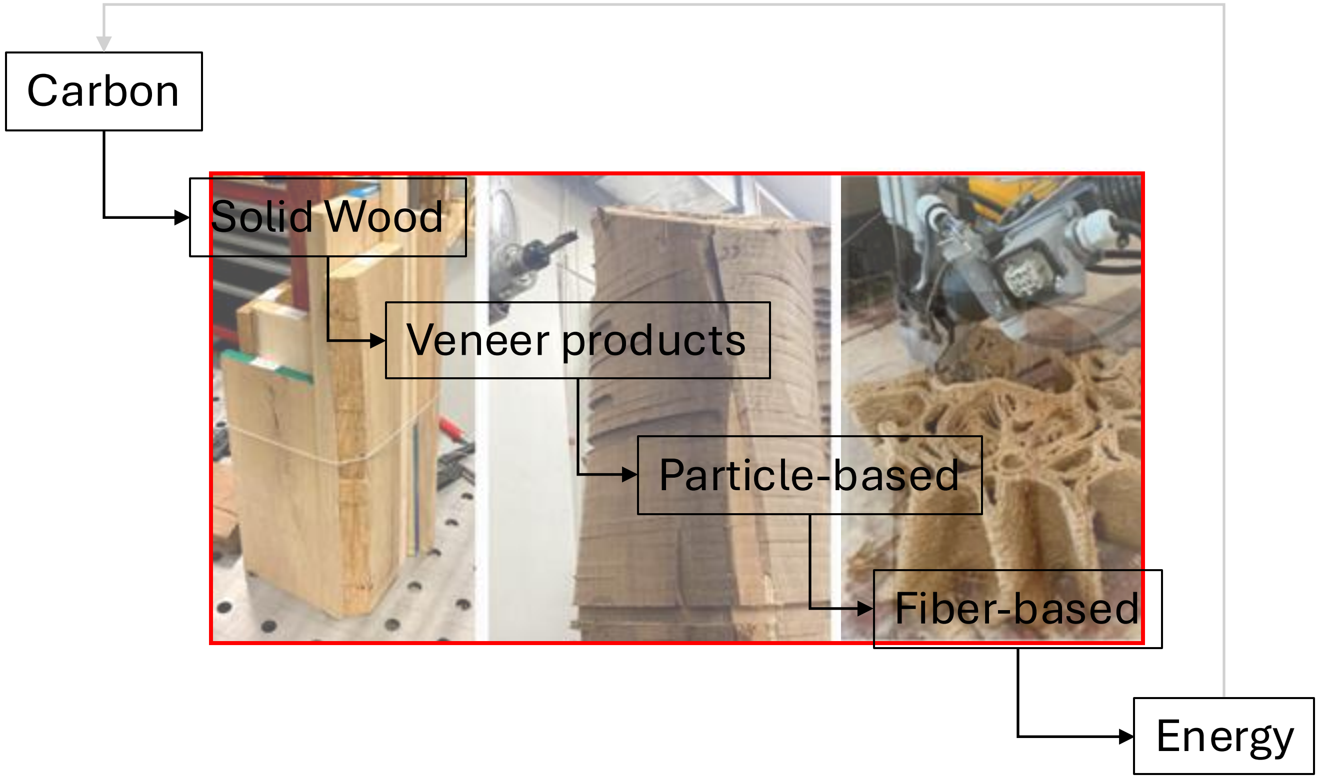
2 Componential Data

Research/on-going



Digital representation of wood packing



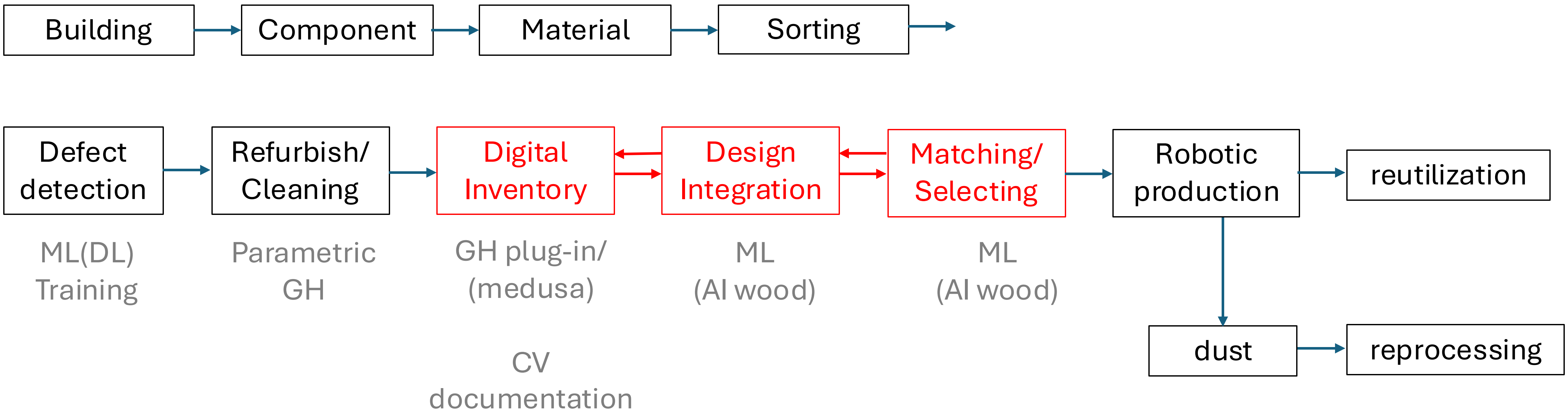


Planning: timber-concrete mix
Conventional: concrete
Timber: timber structure
Light orange: release of carbon in 50 yrs
Orange: embodied carbon

Decarbonization and circular wood: a cascading approach

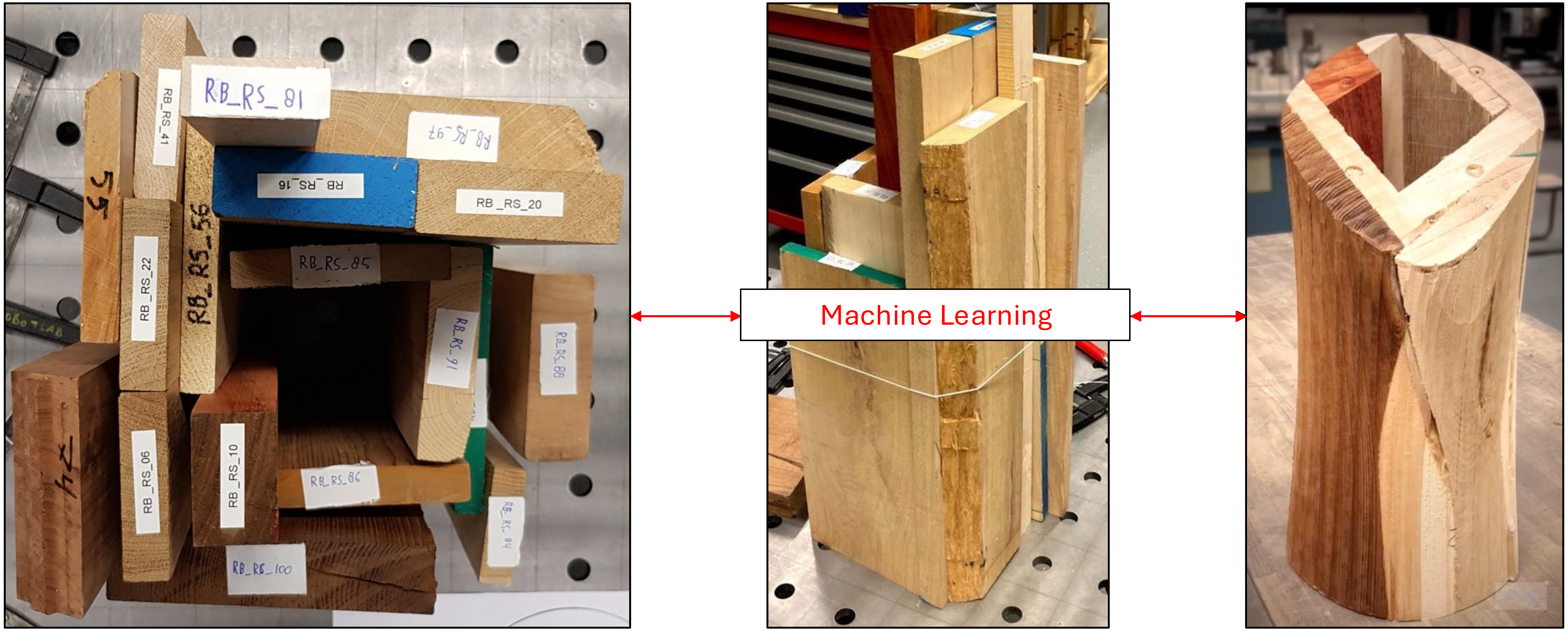
State-of-the-Art

(Inter Höglmeier et al., 2017; Klinge et al., 2023; Kuzman et al., 2024)



Digital workflow for circular design

State-of-the-Art

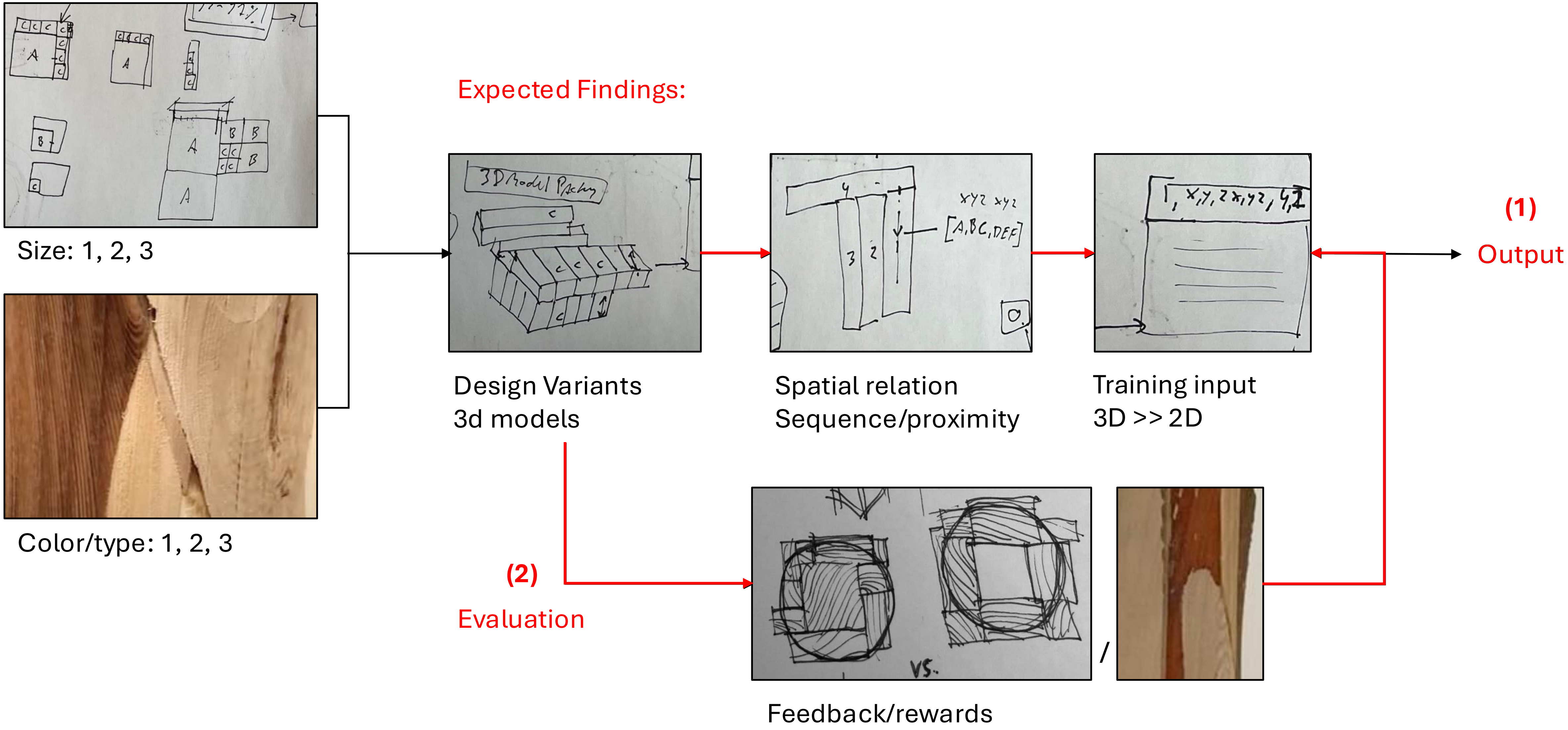


Wood Stock Inventory   Assemblies   Target

AI-supported Design-to-Construction Workflow: Circular Wood In The Neighborhood

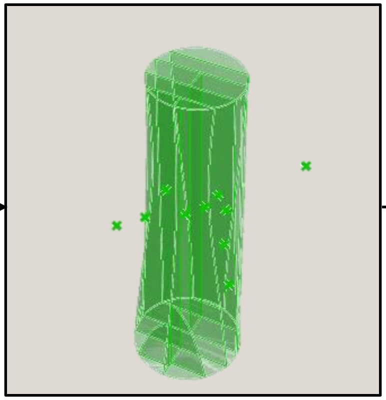
Approach

(@Robotic Building, 2022)

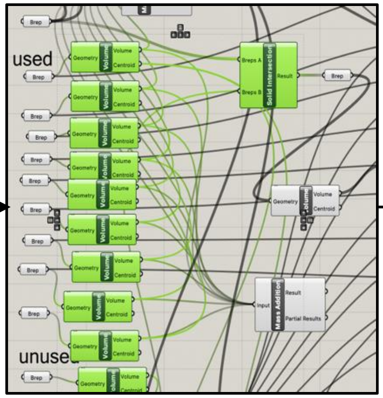


Research Design and Objective

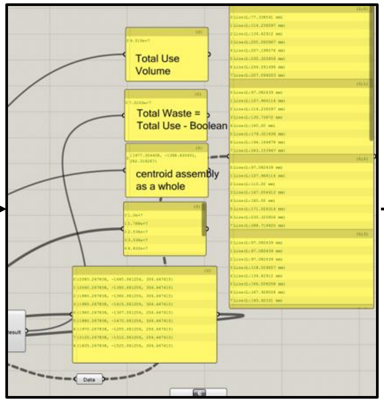
Physical design outcome



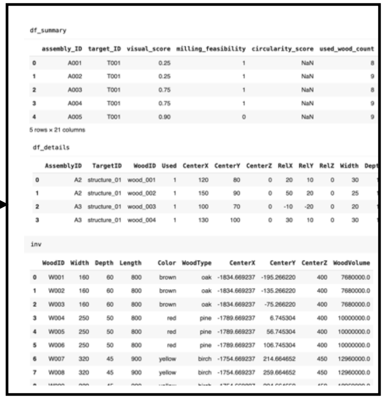
Digital 3D representation



Transcoding: geo-metrical relation



Panels: flattened structured data



CSVs: input for python/AI

Visual/Spatial → Text-based

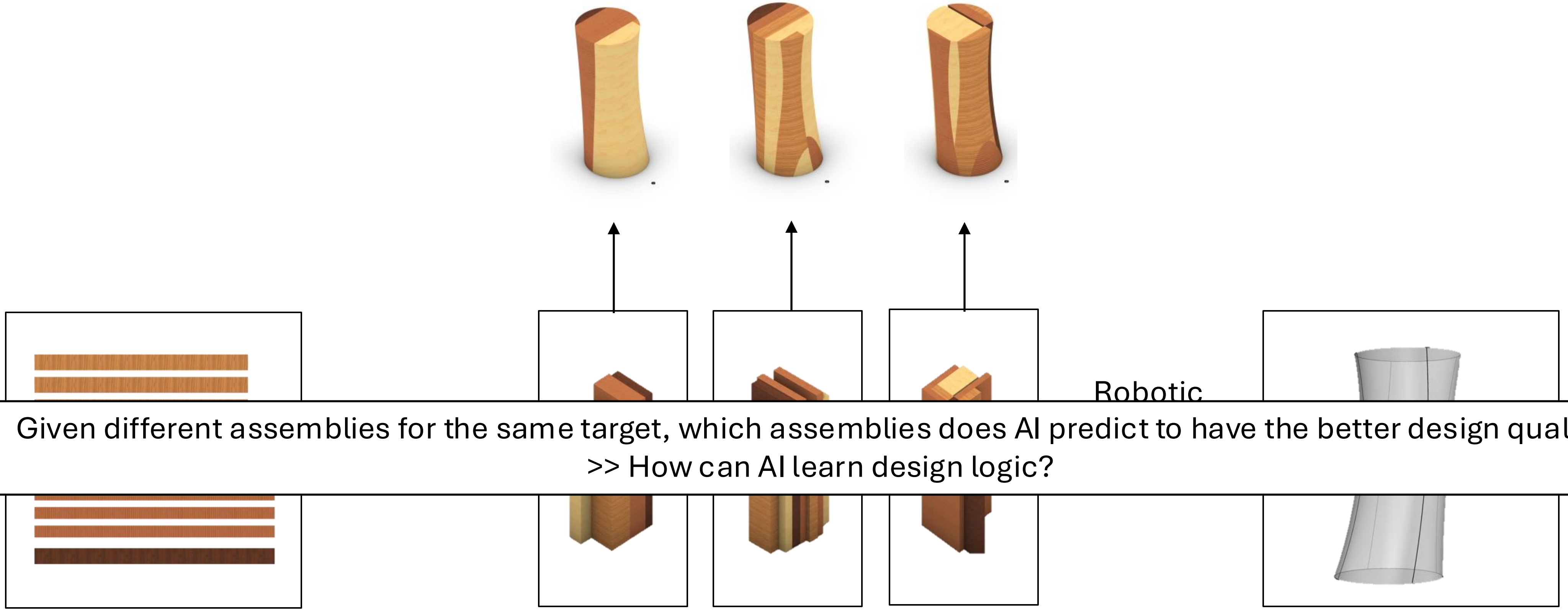
(Rhino/Grasshopper >> ChatGPT/Google Collab)

In parallel: visual/spatial and latent environment

Methodology

Research Aim

Understand (Non-)Spatial Association within Design Assemblies

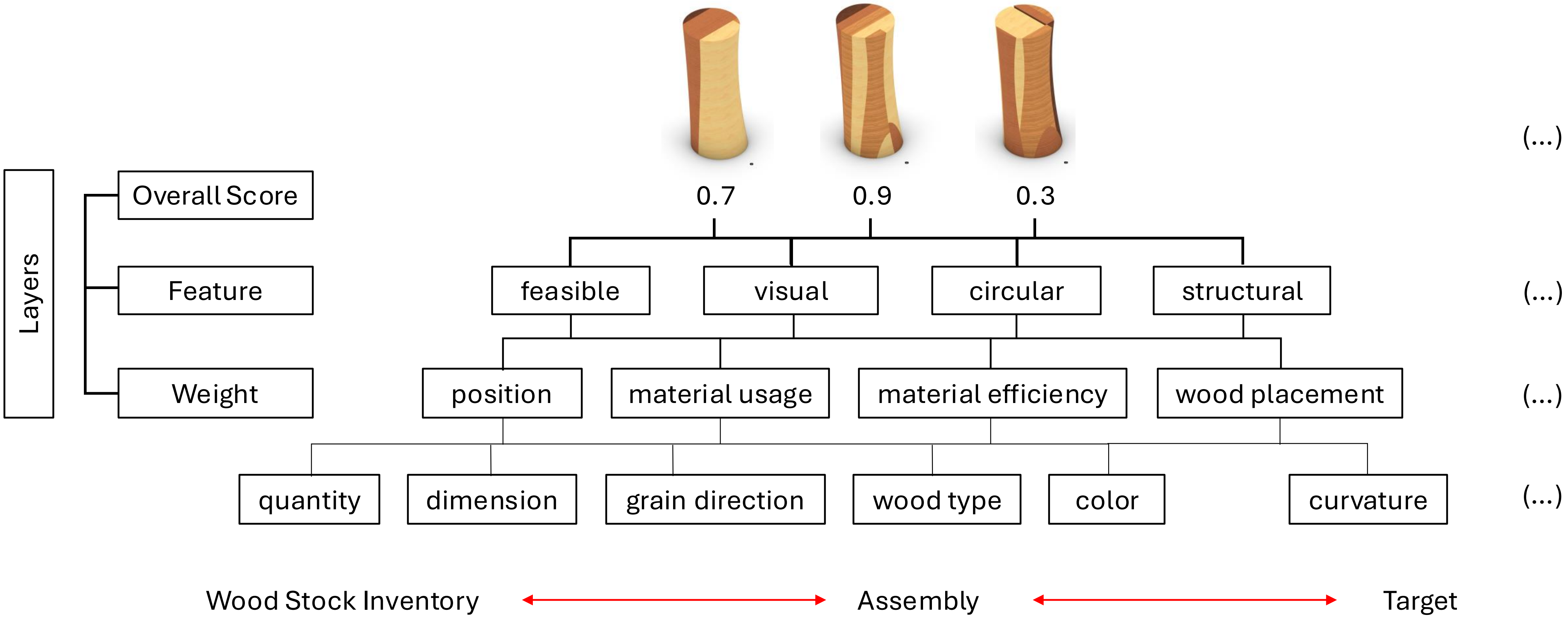


Given different assemblies for the same target, which assemblies does AI predict to have the better design quality?
>> How can AI learn design logic?

Wood Stock Inventory
(10 pieces)

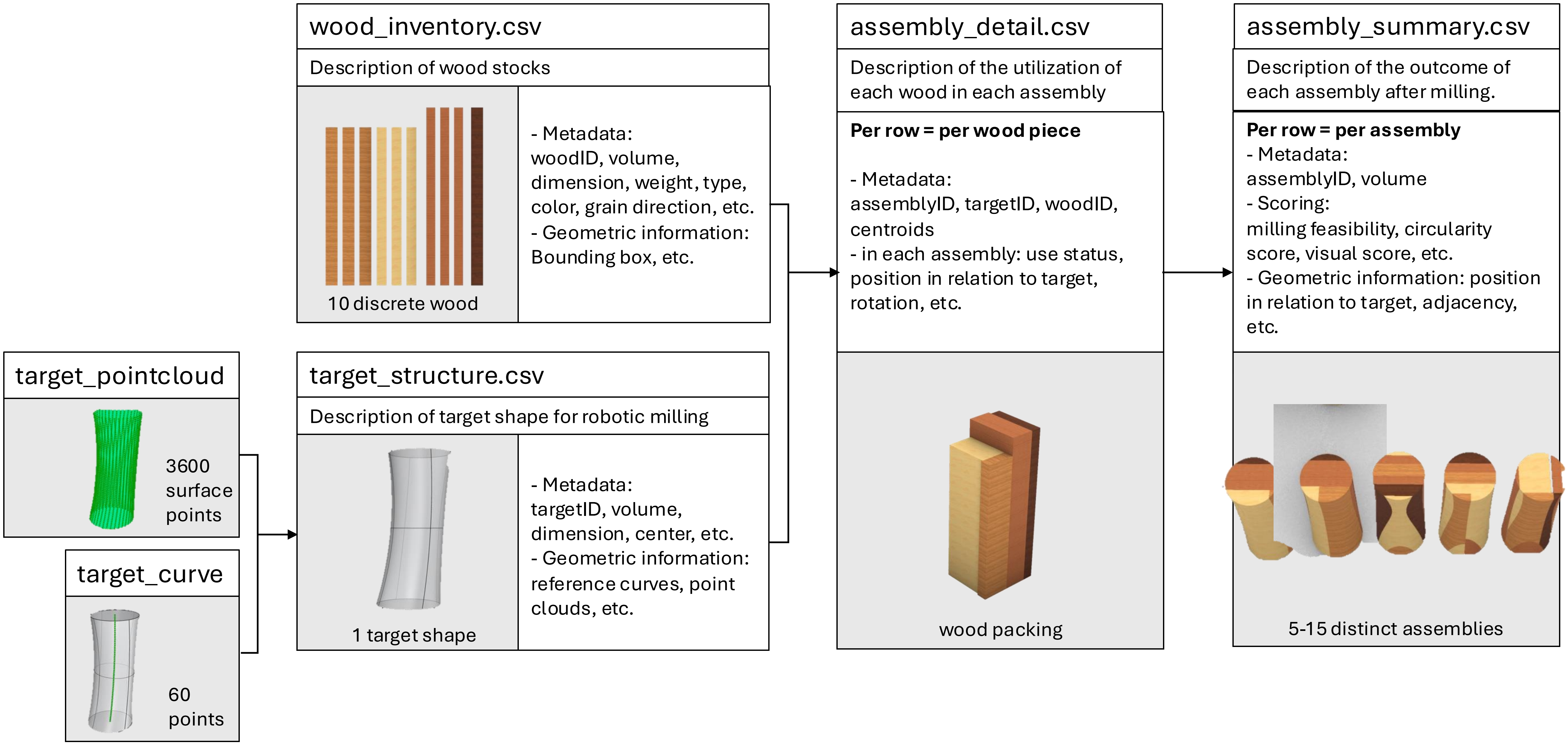
Assemblies
(5 assemblies)

Target
(1 target shape)



Deep learning and neural network

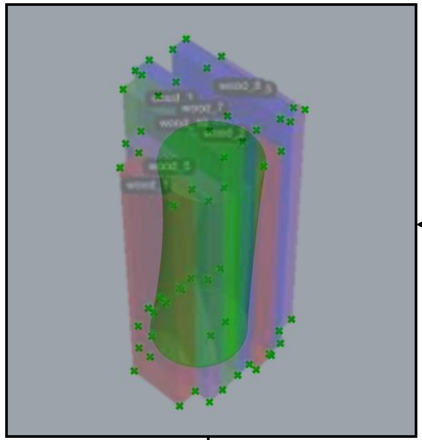
Method



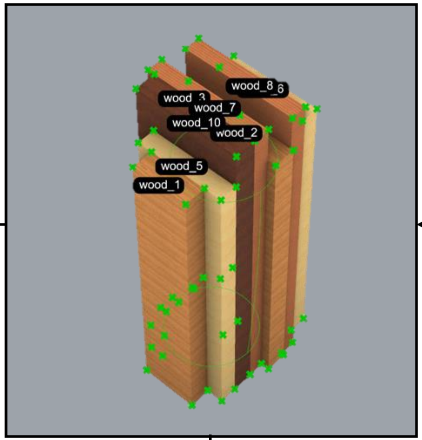
Dataset and Schema

Training Input and Output

Translation/transcoding of componential information



Placement



Assembly

Quality

assembly_ID	target_ID	wood_ID	used	C0_X	C0_Y	C0_Z	C1_X
A001	T001	W001	0				
A001	T001	W002	0				
A001	T001	W003	1	-1040.0978	-1040.097797	-1040.097798	-1040.097799
A001	T001	W004	1	-1050.0978	-1284.918452	350	-1200.097796
A001	T001	W005	1	-1050.0978	-1329.918452	350	-1200.097796
A001	T001	W006	1	-1050.0978	-1374.918452	350	-1200.097796
A001	T001	W007	1	-920.097796	-1477.418452	300	-1210.097796
A001	T001	W008	1	-1075.0978	-1472.418452	300	-1210.097796
A001	T001	W009	1	-1075.0978	-1522.418452	300	-1210.097796
A001	T001	W010	1	-1075.0978	-1422.418452	300	-1190.097796
A002	T001	W001	1	-145.097796	-1522.418452	300	49.902204
A002	T001	W002	1	-40.097796	-1477.418452	300	-10.097796
A002	T001	W003	1	19.902204	-1477.418452	300	-70.097796
A002	T001	W004	0				
A002	T001	W005	1	-40.097796	-1232.418452	350	-120.097796
A002	T001	W006	1	-50.097796	-1284.918452	350	-170.097796
A002	T001	W007	1	-50.097796	-1329.918452	350	-210.097796
A002	T001	W008	1	-50.097796	-1374.918452	350	-210.097796
A002	T001	W009	1	79.902204	-1477.418452	300	-210.097796
A002	T001	W010	1	-95.097796	-1522.418452	300	-190.097796
A003	T001	W001	1				881.116653
A003	T001	W002	1				954.171701

Coordinates

used_wood_count	unused_wood_count
8.0	2.0
9.0	1.0
8.0	2.0
9.0	1.0
9.0	1.0

used_wood_volume	waste_wood_volume
92760000.0	67843000.0
98120000.0	73200000.0
85160000.0	60243000.0
95160000.0	70243000.0
100440000.0	76512000.0

pretrim_wood_volume	posttrim_wood_volume
160603000.0	24917000.0
171320000.0	24917000.0
145403000.0	24917000.0
165403000.0	24917000.0
176952000.0	23928000.0

Pre-computed features

	assembly_ID	target_ID
0	A001	T001
1	A002	T001
2	A003	T001
3	A004	T001
4	A005	T001

visual_score	milling_feasibility
0.25	1.0
0.25	1.0
0.75	1.0
0.75	1.0
0.90	0.0

circularity_score
0.387159
0.365188
0.392595
0.366842
0.361925

Score

$$\textbf{circularity\_score} = \boxed{\text{material\_efficiency}} \times \boxed{e^{-\alpha \cdot N_{\text{used}}}}$$

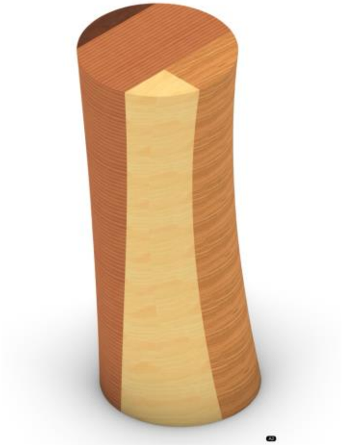
assembly_summary.csv

Y (output) circularity_score

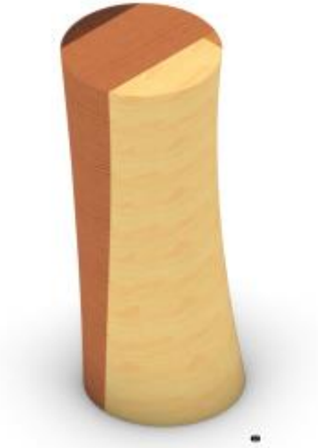
X (input)

	visual_score	milling_feasibility	used_wood_count	unused_wood_count	used_wood_volume	waste_wood_volume	pretrim_wood_volume	posttrim_wood_volume	rel_X	rel_Y	rel_Z	rel_dist
0	0.25	1.0	8.0	2.0	92760000.0	67843000.0	160603000.0	24917000.0	-5.094741	-1.232805	29.870649	30.327081
1	0.25	1.0	9.0	1.0	98120000.0	73200000.0	171320000.0	24917000.0	86.856718	-17.736956	28.248800	93.041301
2	0.75	1.0	8.0	2.0	85160000.0	60243000.0	145403000.0	24917000.0	-8.207530	1.353125	30.782349	31.886478
3	0.75	1.0	9.0	1.0	95160000.0	70243000.0	165403000.0	24917000.0	17.883142	-28.879282	-17.502605	38.212051
4	0.90	0.0	9.0	1.0	100440000.0	76512000.0	176952000.0	23928000.0	28.258722	-10.763132	-27.238227	40.697929

A001
0.387159



A002
0.365188



A003
0.392595

Ground Truth

A004
0.366842

A005
0.361925

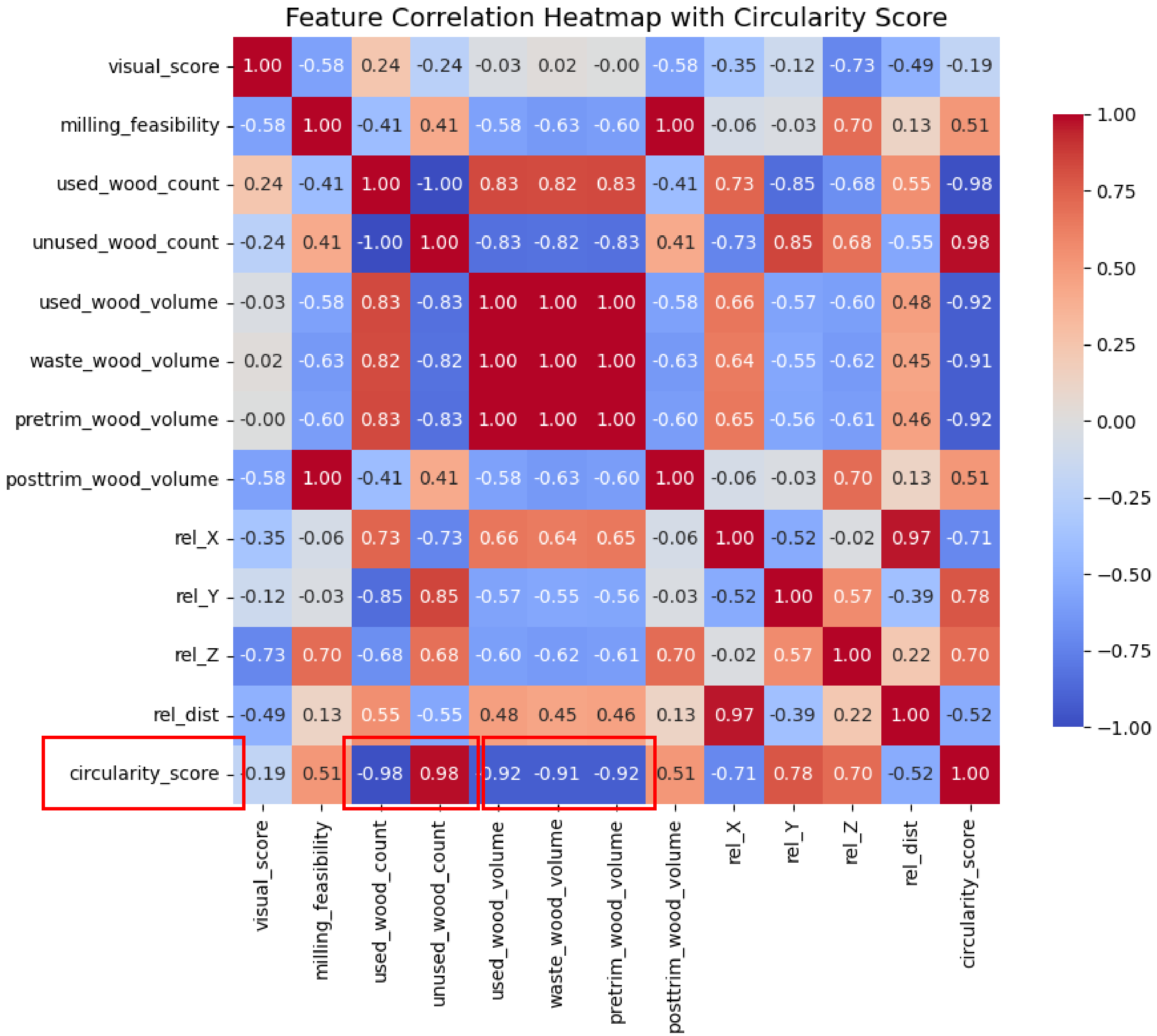
Which is the most circular assembly?

Training_V1

2	5	1	3	4
0.383917	0.348795	0.393716	0.368535	0.363053

Prediction	2	4	1	3	5

Result_V1



Correlation

Result_V1

wood.ipynb

File Edit View Insert Runtime Tools Help

mmands + Code + Text ▷ Run all

```
            "circ": float(row["circularity_score"]),
            "vis":  float(row["visual_score"])
        }

print("Example labels:")
for k in list(labels.keys())[:3]:
    print(k, labels[k])
```

```
Example labels:
A001 {'feas': 1.0, 'circ': nan, 'vis': 0.25}
A002 {'feas': 1.0, 'circ': nan, 'vis': 0.25}
A003 {'feas': 1.0, 'circ': nan, 'vis': 0.75}
```

```
# STEP 4 — Combine into DeepSets dataset format
# ---------------------------------------------
assemblies = []   # List of (wood_piece_matrix, label_dict)

for a_id in assembly_groups:
    assemblies.append((assembly_groups[a_id], labels[a_id]))

print("DeepSets dataset assembled.")
print("Total assemblies =", len(assemblies))
```

```
DeepSets dataset assembled.
Total assemblies = 5
```

```
X = df[coord_cols].values

y_feas = df["milling_feasibility"].values      # classification
y_circ = df["circularity_score"].values        # regression
y_vis  = df["visual_score"].values             # regression
```

	assembly_ID	target_ID	wood_ID	used	C0_X	C0_Y	C0_Z	C1_X	C1_Y	C1_Z
0	A001	T001	W001	0	NaN	NaN	NaN	NaN	NaN	NaN
1	A001	T001	W002	0	NaN	NaN	NaN	NaN	NaN	NaN
2	A001	T001	W003	1	0.480938	329.858872	-1339.916670	0.480935	329.858869	-1339.916673
3	A001	T001	W004	1	-9.519062	85.038217	50.181128	-159.519062	-177.461783	-399.818872
4	A001	T001	W005	1	-9.519062	40.038217	50.181128	-159.519062	-127.461783	-399.818872

5 rows × 41 columns

```
# STEP 1 — Detect coordinate feature columns
# ---------------------------------------------

coord_cols = [
    c for c in detail.columns
    if (c.startswith("C") and c[1].isdigit() and c[2] == "_")
]

print(len(coord_cols), "feature columns:", coord_cols)
```

```
27 feature columns: ['C0_X', 'C0_Y', 'C0_Z', 'C1_X', 'C1_Y', 'C1_Z', 'C2_X', 'C2_Y', 'C2_Z', 'C3_X', 'C3_Y'
```

```
# STEP 2 — Group wood pieces by assembly
# ---------------------------------------------

assembly_groups = {}
```

Coordinates as input

Training_V2