# Computational Design and Fabrication of Structures and Mechanisms

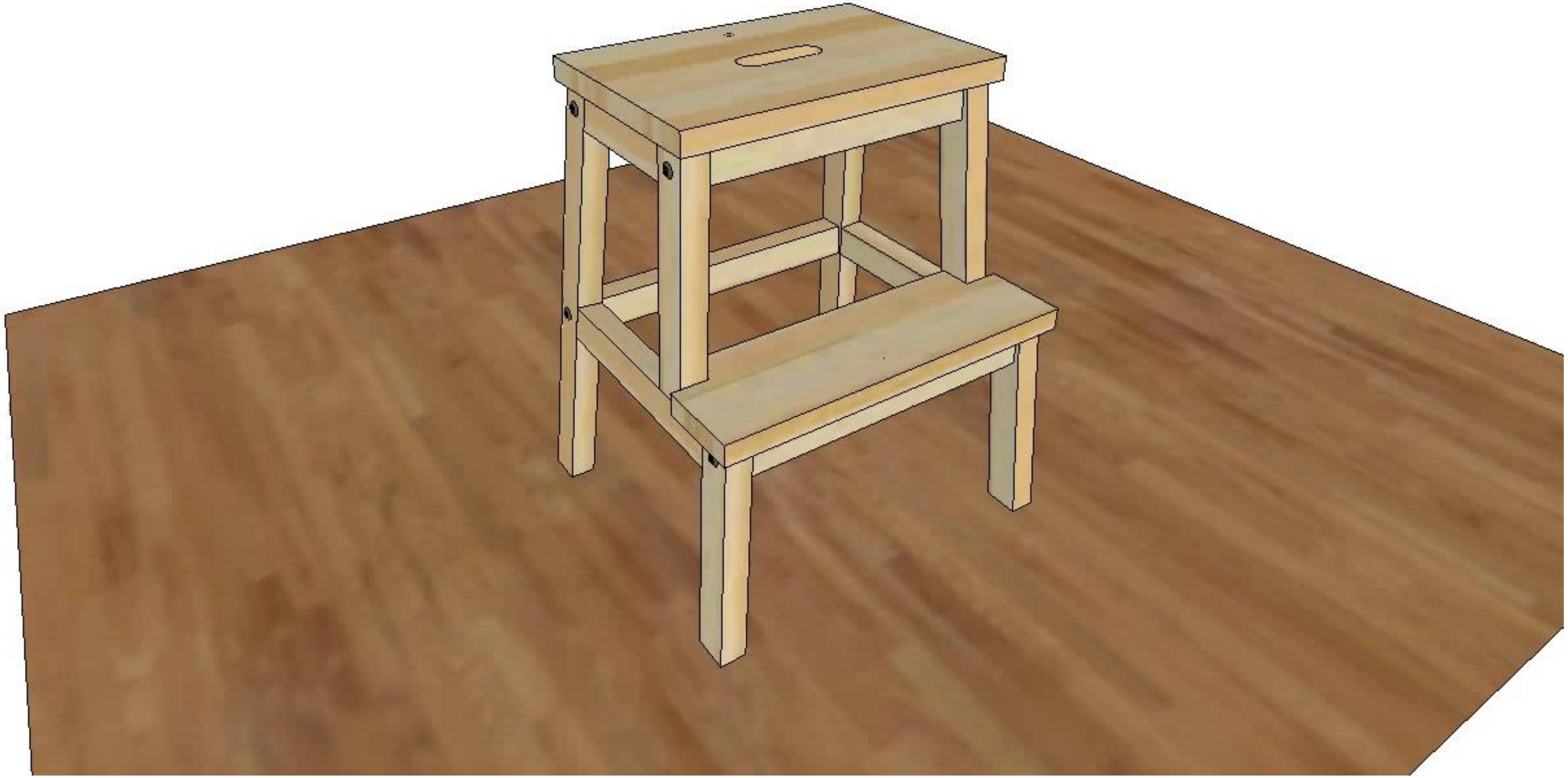
Peng Song

2017.12.28

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

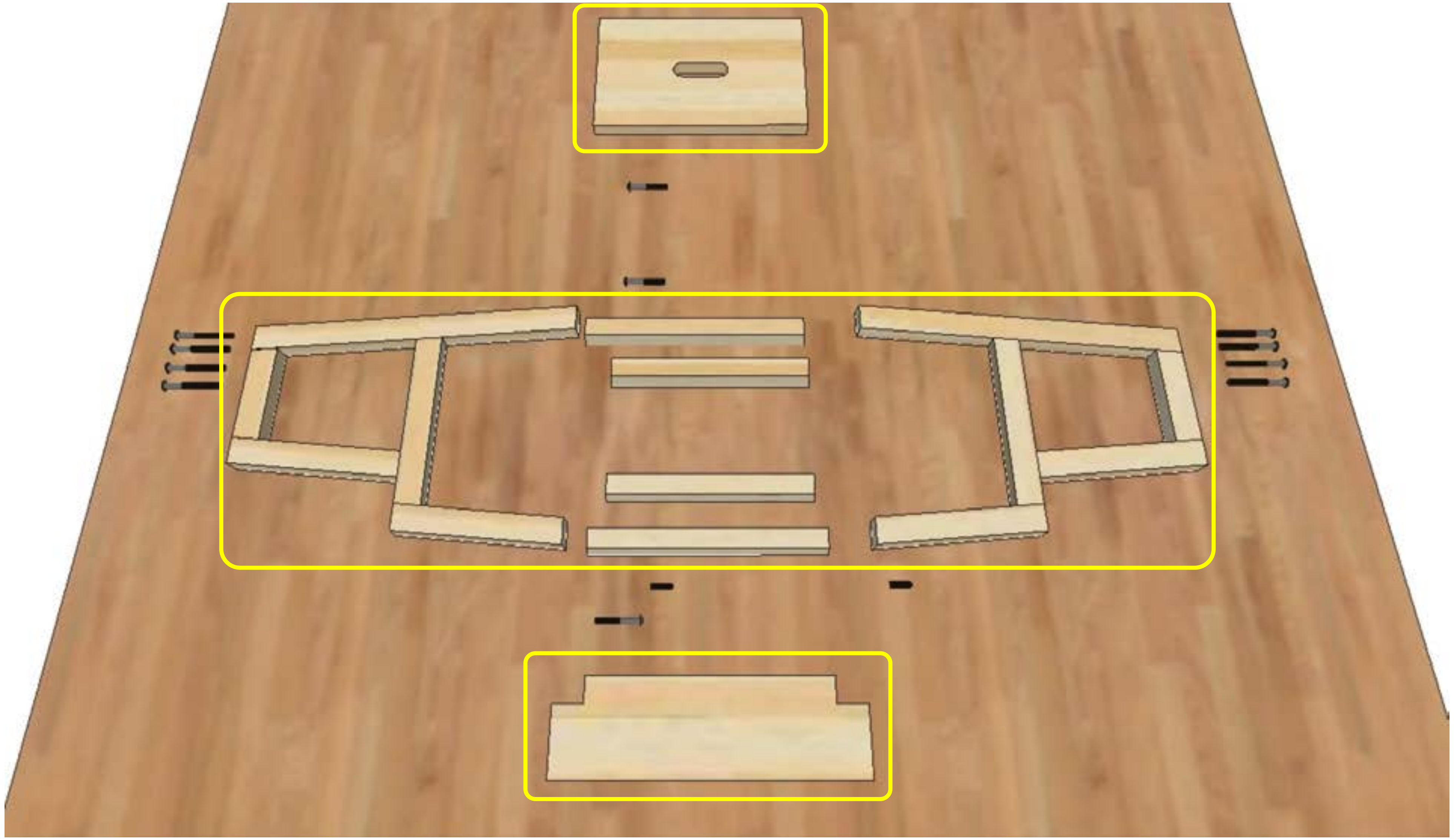
# 3D Assembly

# 3D Assembly



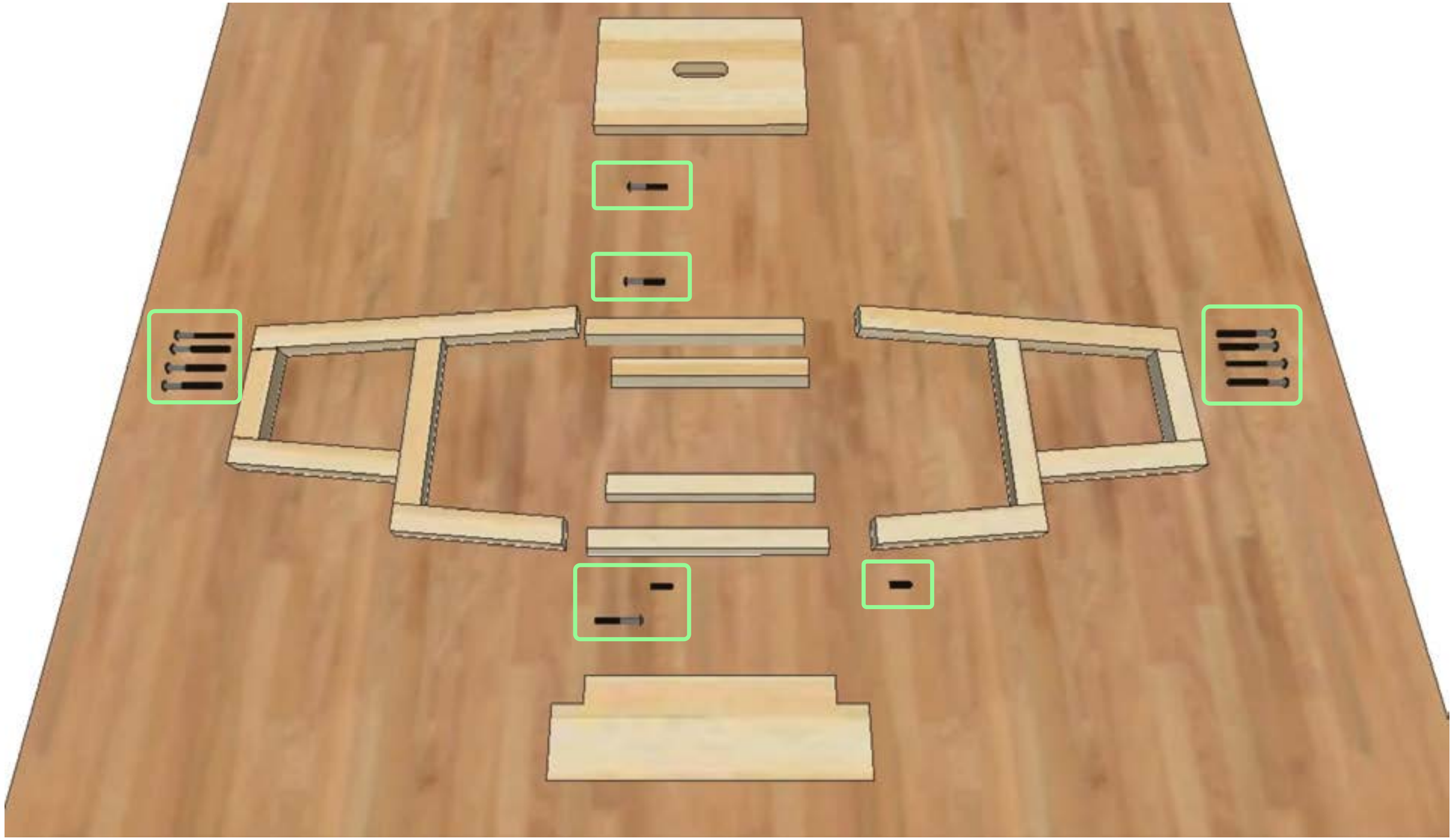
# **3D Assembly:** Components

- Parts



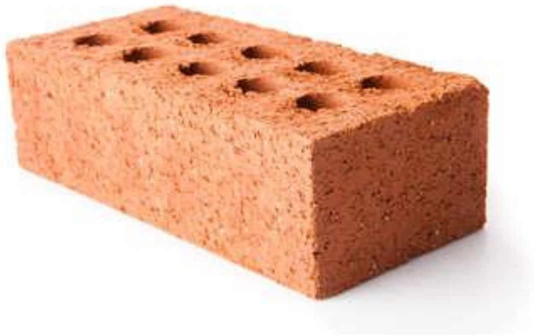
# **3D Assembly:** Components

- Parts
- Connectors



# **3D Assembly:** Advantages

#1 Build huge objects



Part

Connector

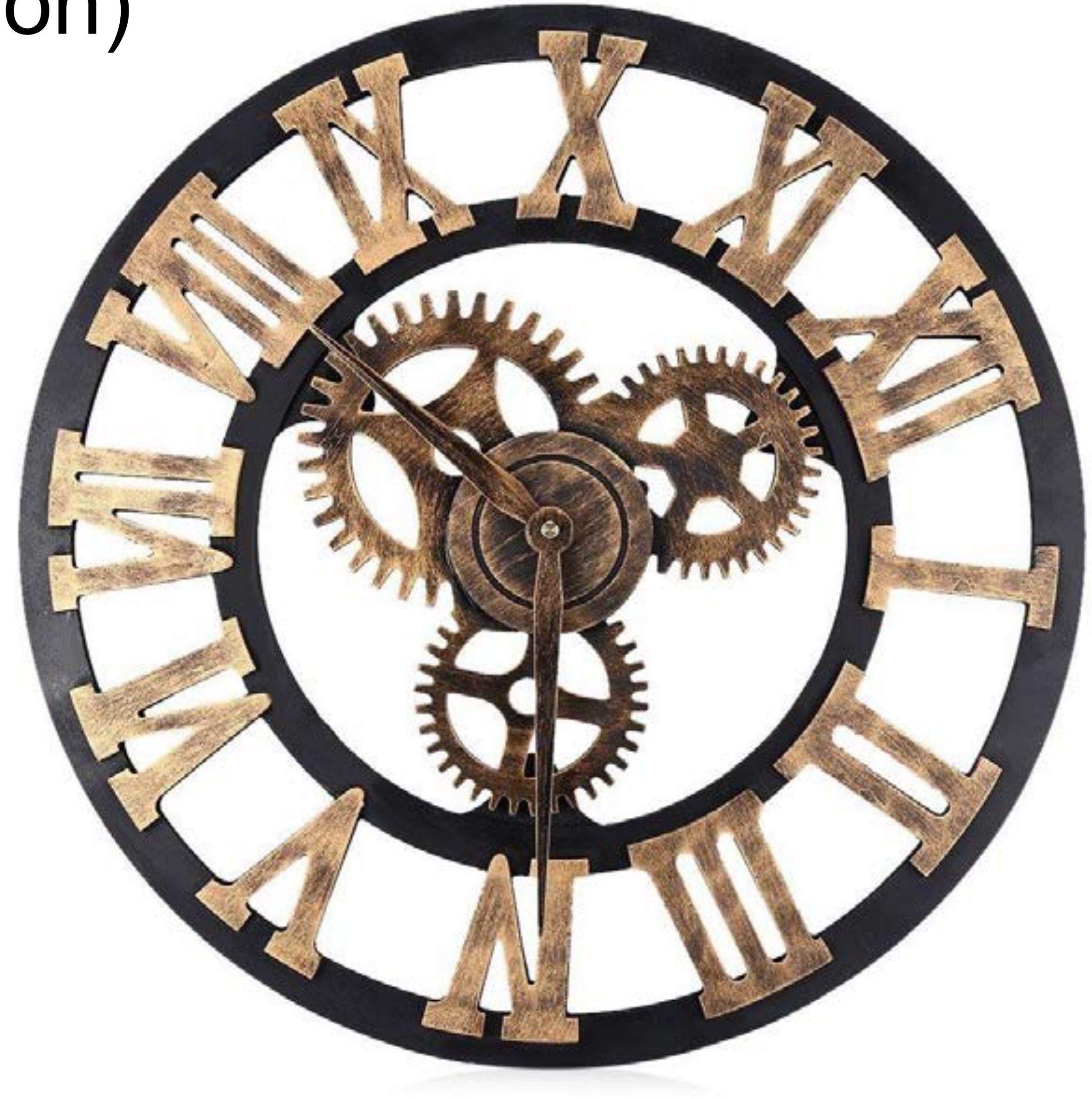
# **3D Assembly:** Advantages

#2 Easy for storage, transport

disassemble and pack

# **3D Assembly:** Advantages

#3 Functionality (relative part motion)



# 3D Assembly

Structure





Mechanism




# **3D Assembly:** Structure

- **Task**: bear load
- **DOF**: = 0
- **Connector**: glue, nail, screw, etc.





# **3D Assembly:** Mechanism

- **Task**: transfer motion
- **DOF**: > 0
- **Connector**: gear, linkage, cam, etc.




# Content

3D Assembly

Structure

Mechanism

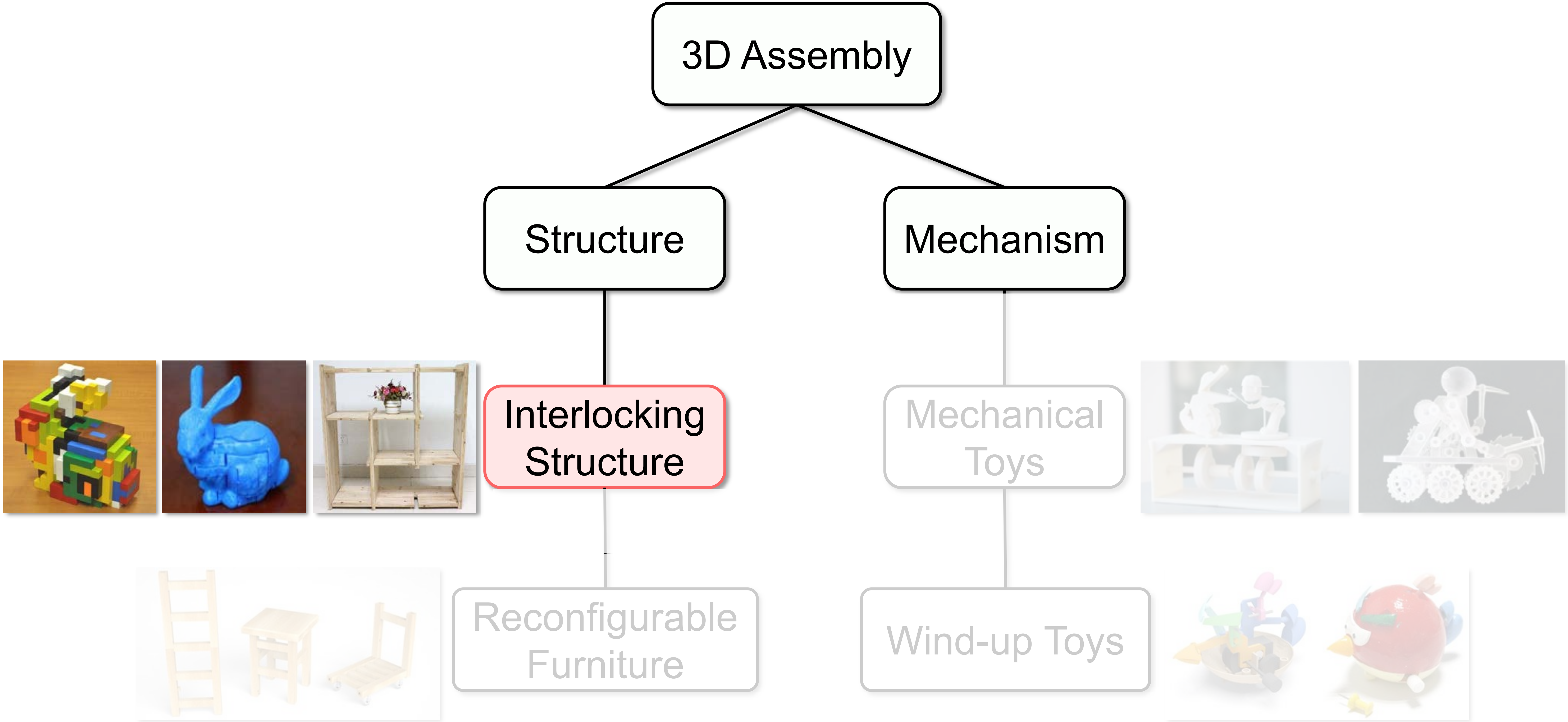
Interlocking Structure

Mechanical Toys

Reconfigurable Furniture

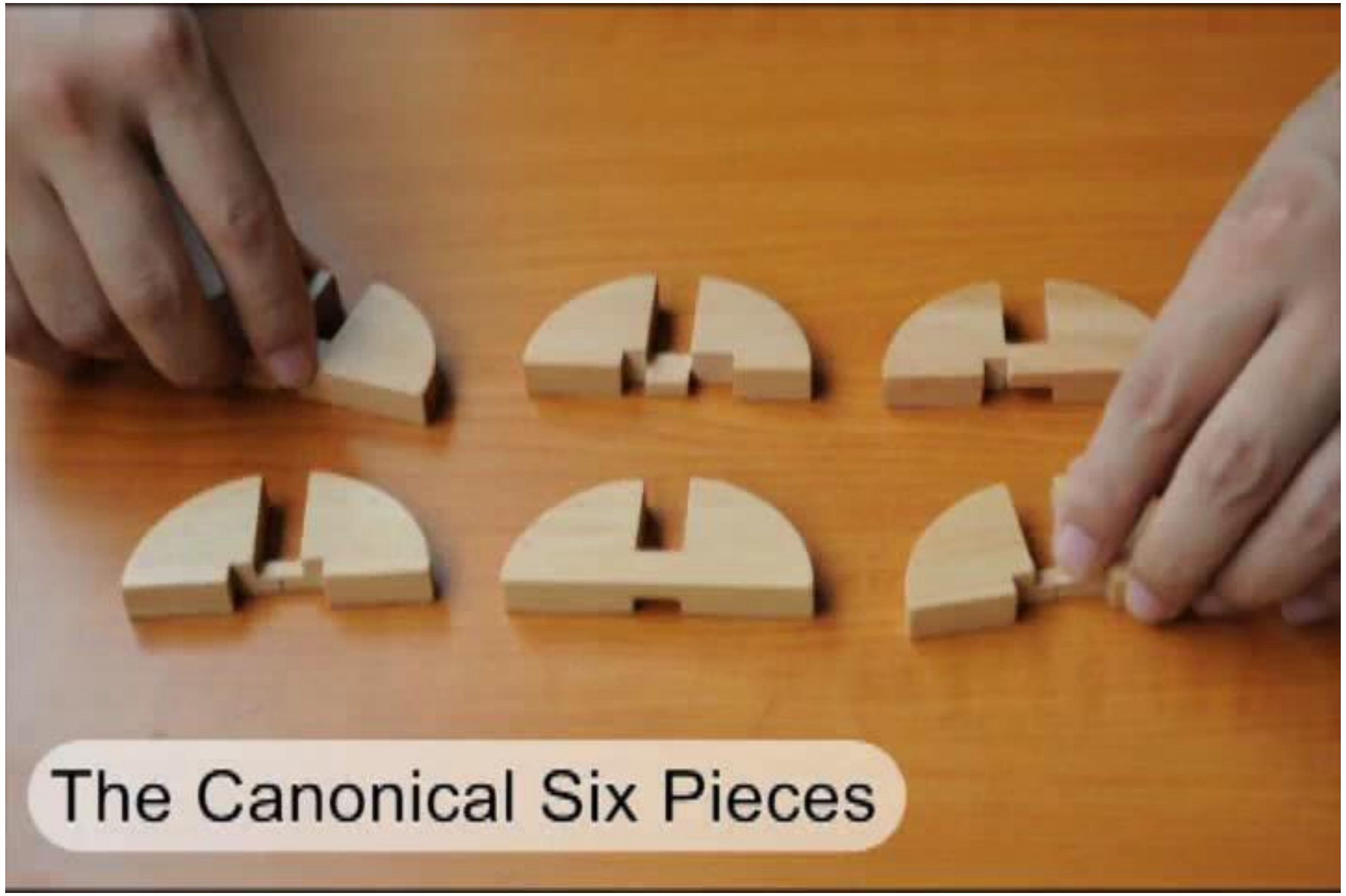
Wind-up Toys

# Overview


3D Assembly
Structure
Mechanism
Interlocking Structure
Mechanical Toys
Reconfigurable Furniture
Wind-up Toys

# Interlocking Structure

Parts are connected (i.e., interlocked) with one another using their own geometry to form a steady assembly, without using any additional connectors



# Characteristics

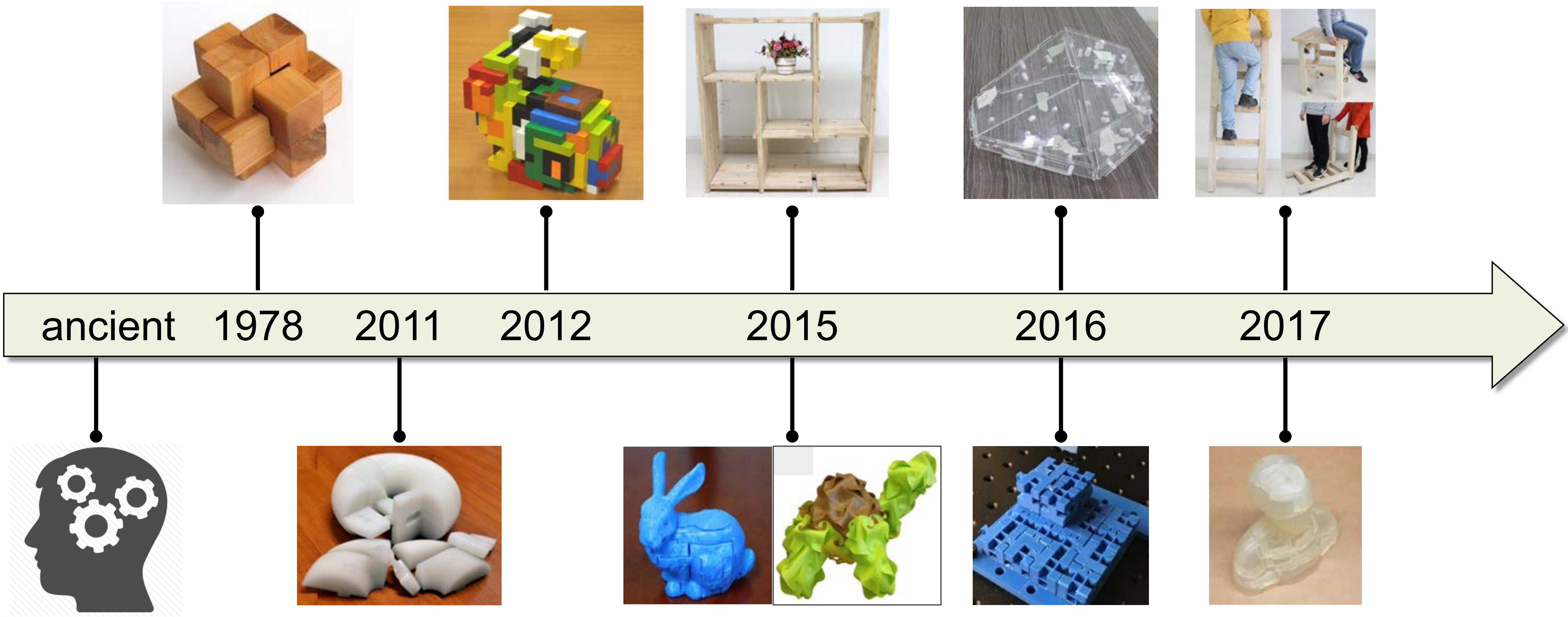
- Parts need to be assembled following a certain order
- Parts can be repeatedly disassembled and reassembled
- No additional connector is required

# Characteristics

| Connection Approach |  Friction |  Glue |  Screw |  Interlocking |
|---|---|---|---|---|
| **Stability** | | | ✓ | ✓ |
| **Disassemblable** | ✓ | | ✓ | ✓ |
| **Avoid connectors** | ✓ | | | ✓ |

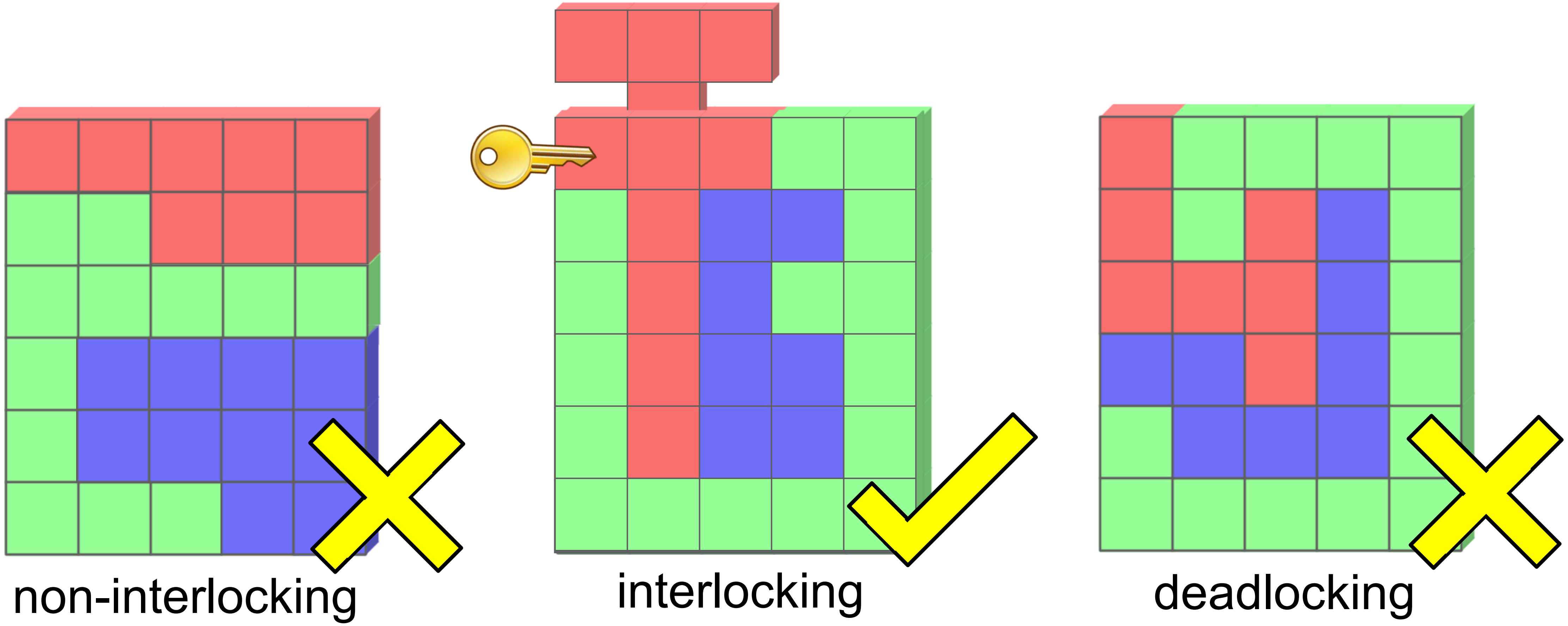
# Applications


# Related Works



# **Formal Definition:** Interlocking Structure

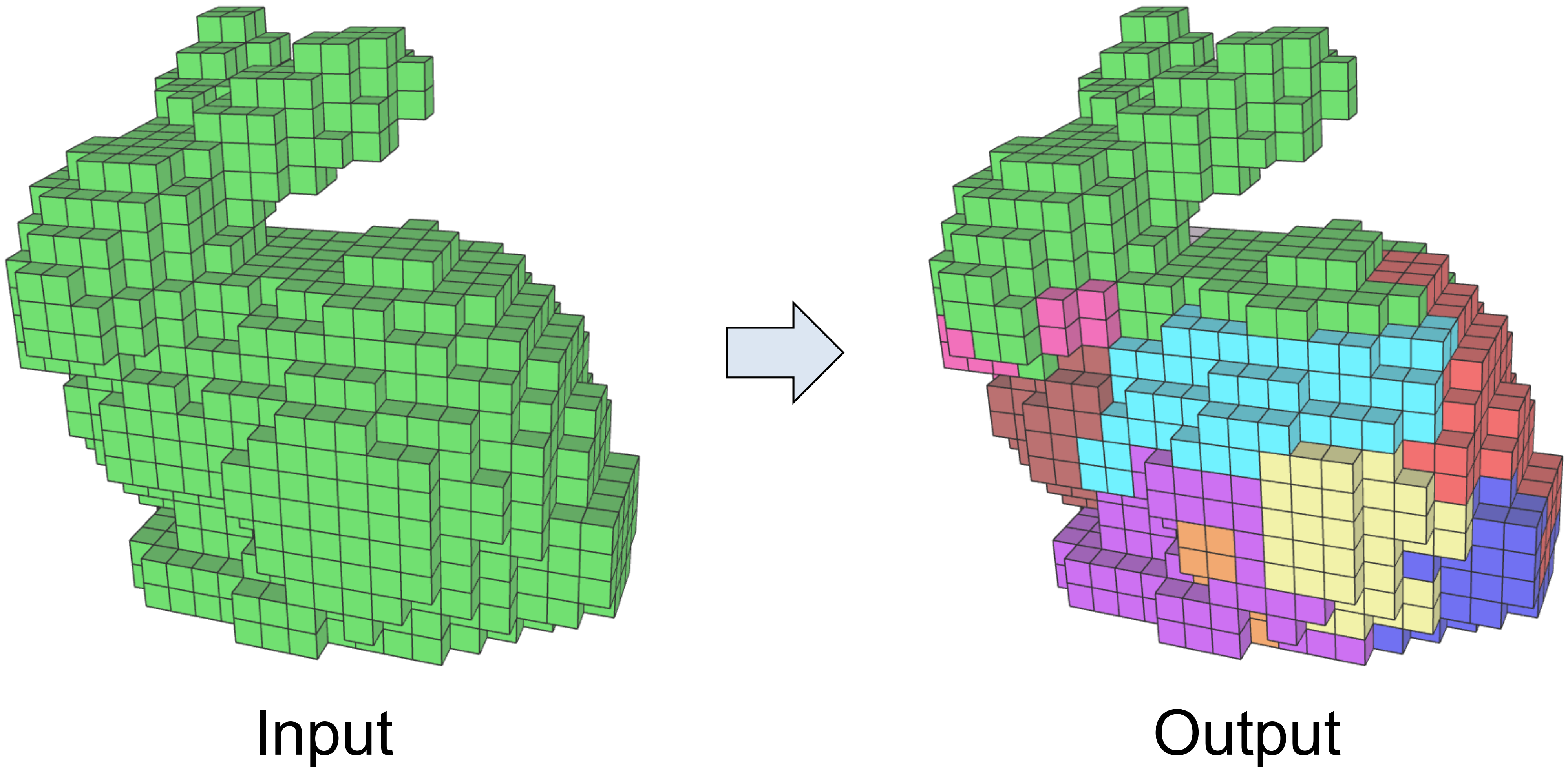
An assembly is interlocking if only one movable part (key), while all other parts, as well as any subset of parts, are immobilized




[Note: these illustrations are in 2D]

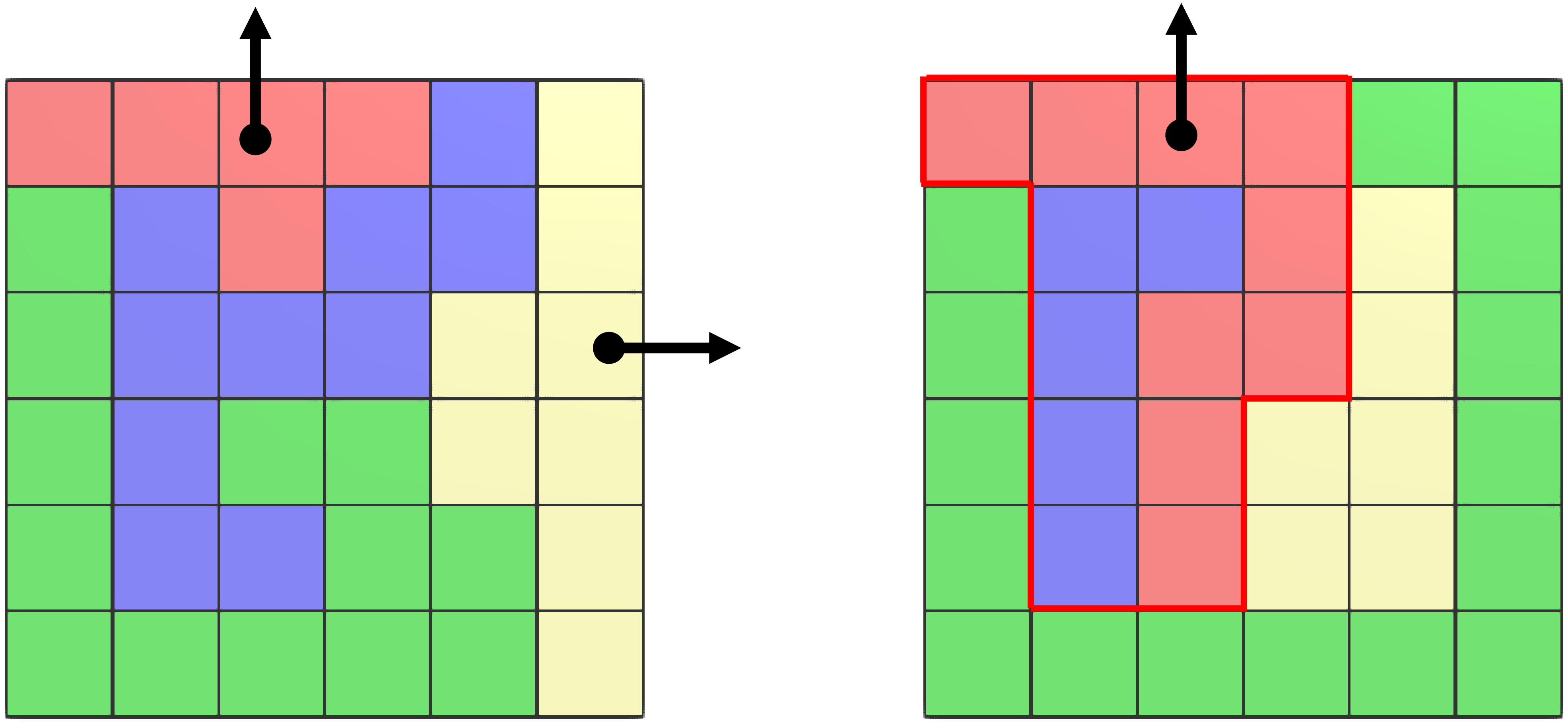
# **Problem:** Design Interlocking Structure

Given a target 3D shape, construct geometry of interlocking parts



# **Challenge #1** Global Interlocking

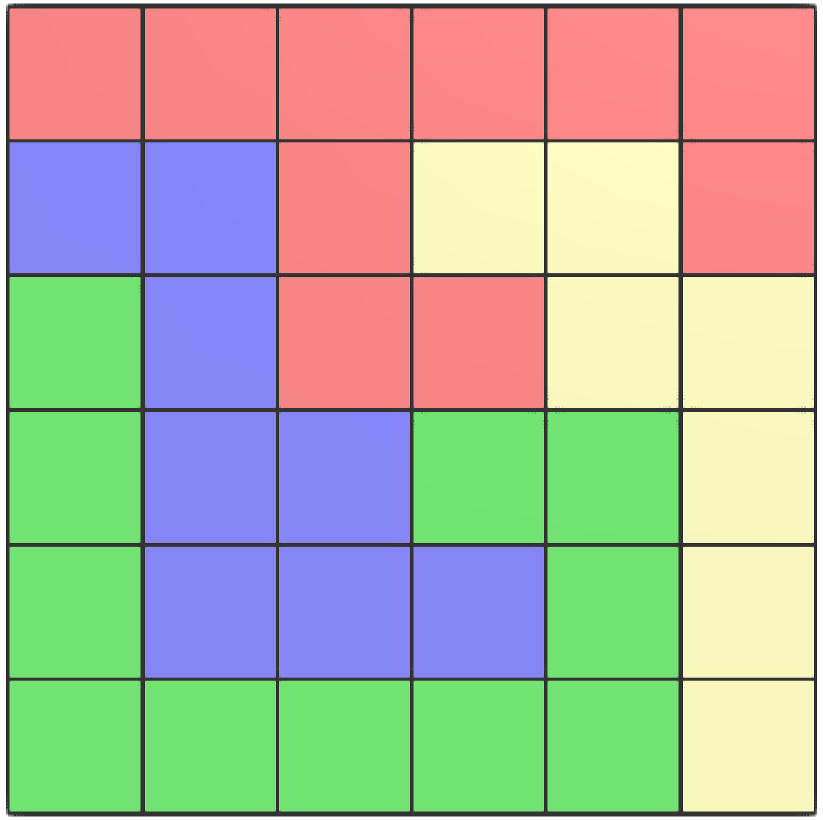
- Need to check immobilization of every single part and every subset of parts
- The complexity is $2^K$, where $K$ is the number of parts



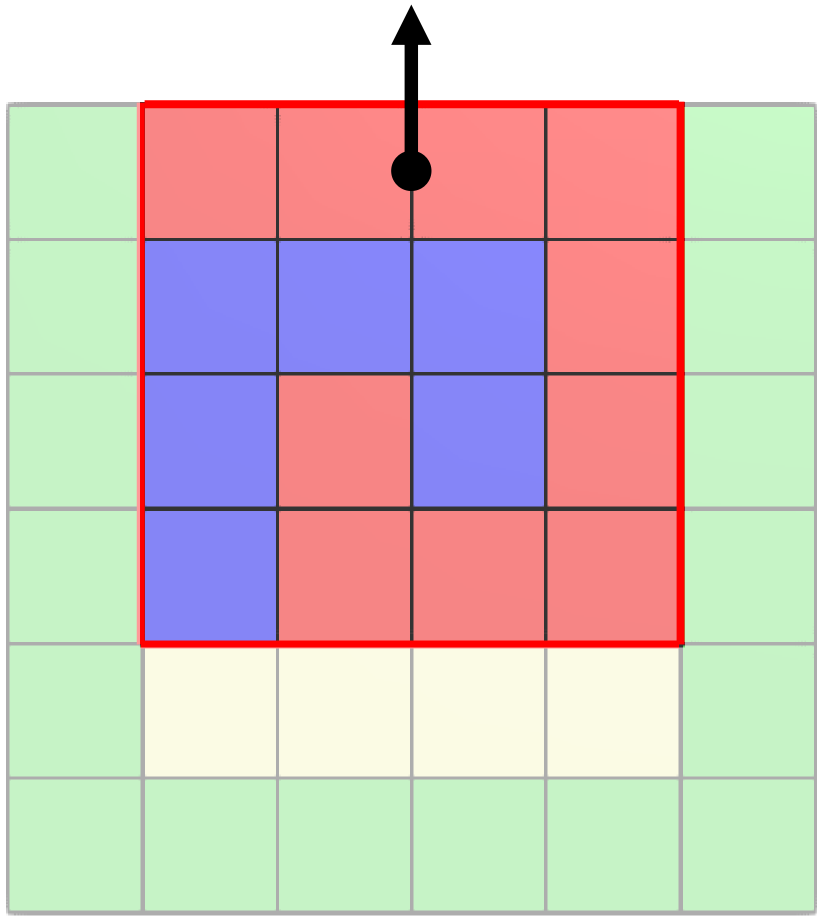
Movement of a single part    Movement of a subset of parts

# Challenge #2  Disassemblable

- The assembly can be progressively disassembled into a set of individual parts following a certain order



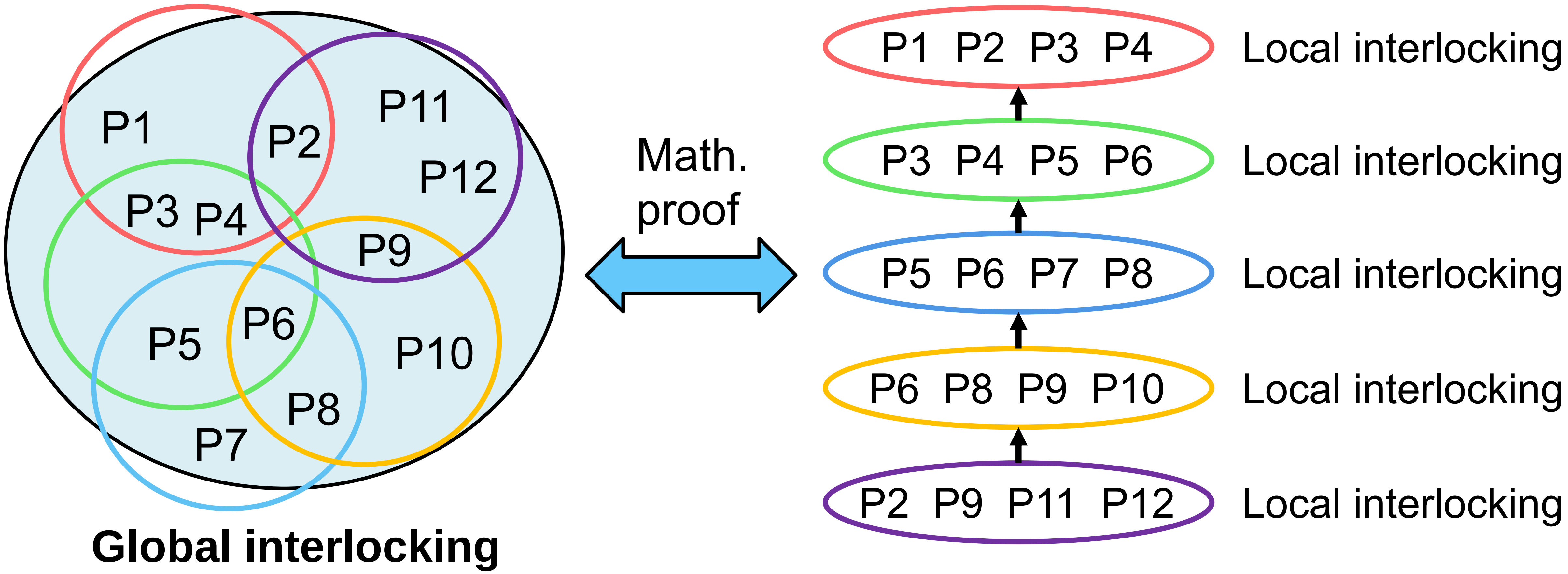
Deadlocking



Local deadlocking

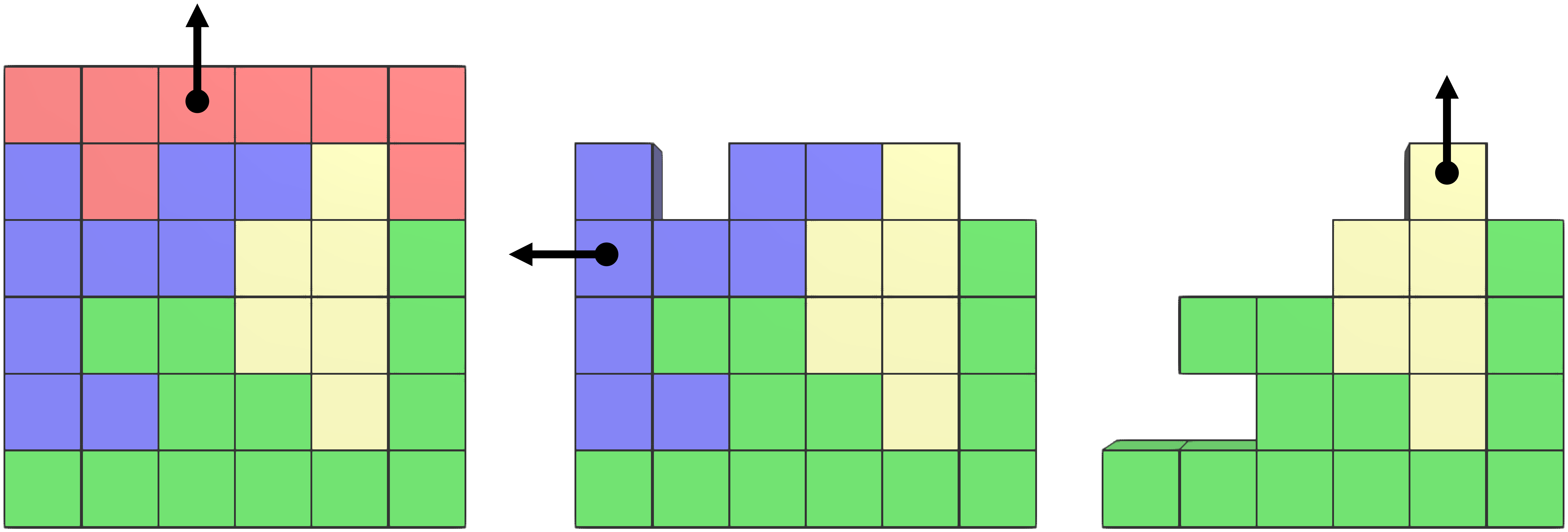
# **Key Idea #1:** Local Interlocking Groups

- Break the set of parts into multiple local groups
- Ensure local interlocking for each group
- Enforce dependency among the local interlocking groups



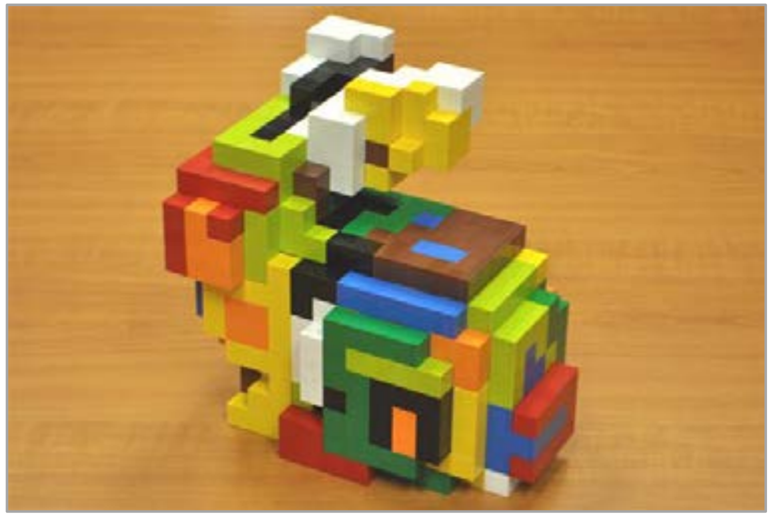
# **Key Idea #2:** Motion-guided Geometry Construction

- Construct parts geometry guided by its target moving direction, which is determined when enforcing the local interlocking constraint

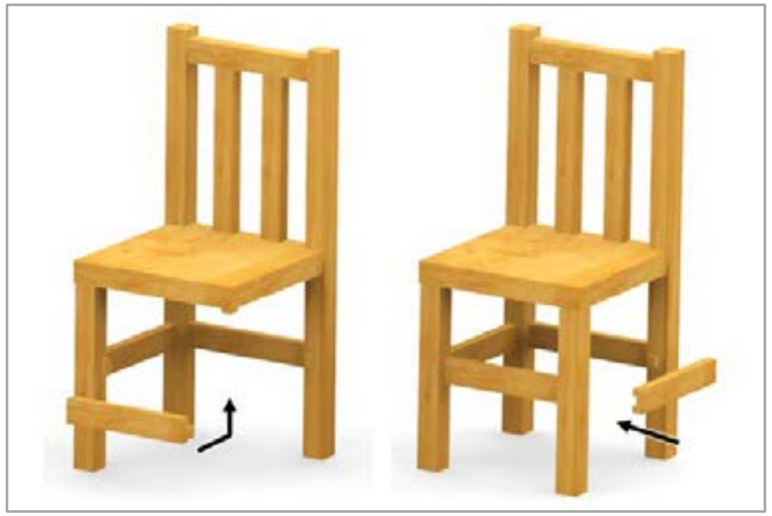


# Four Research Projects

Interlocking Puzzle, 2012



Interlocking Furniture, 2015

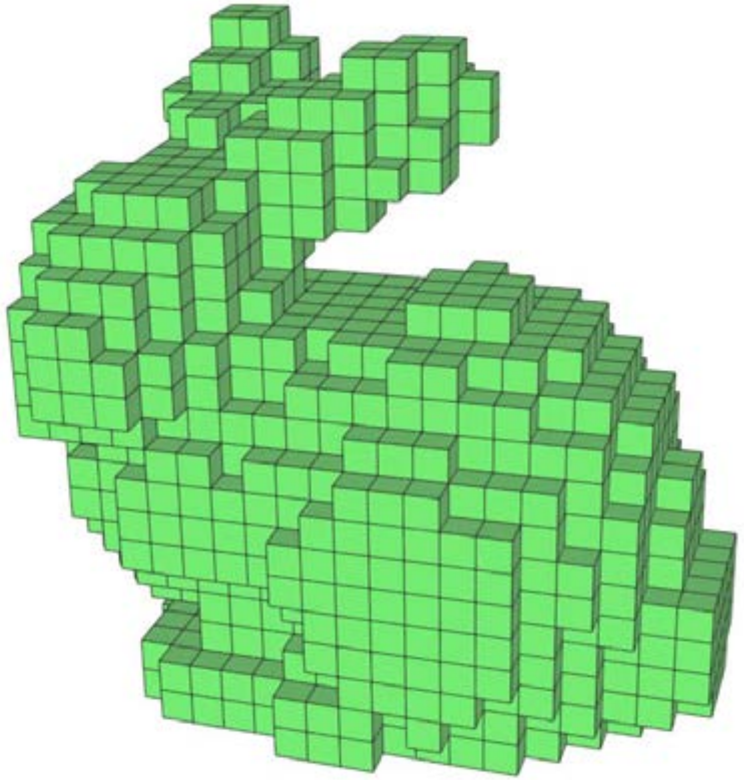


Interlocking Polyhedron, 2016

Interlocking Reconfigurables, 2017

# Recursive Interlocking Puzzles

Song et al.    SIGGRAPH Asia 2012

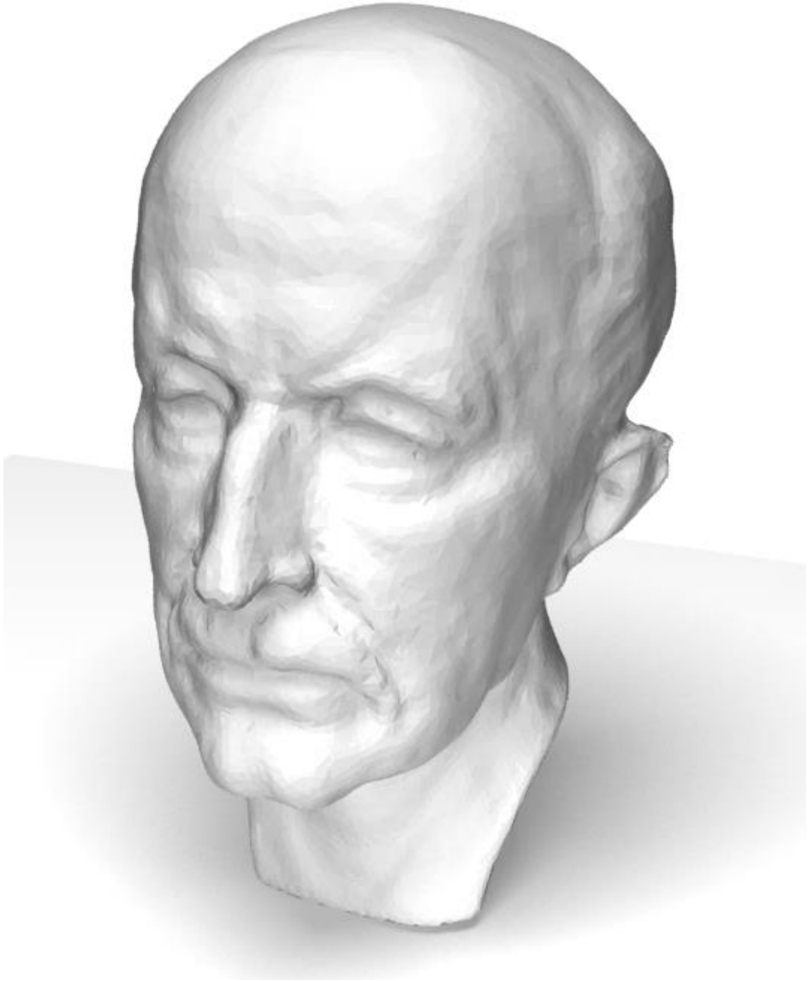


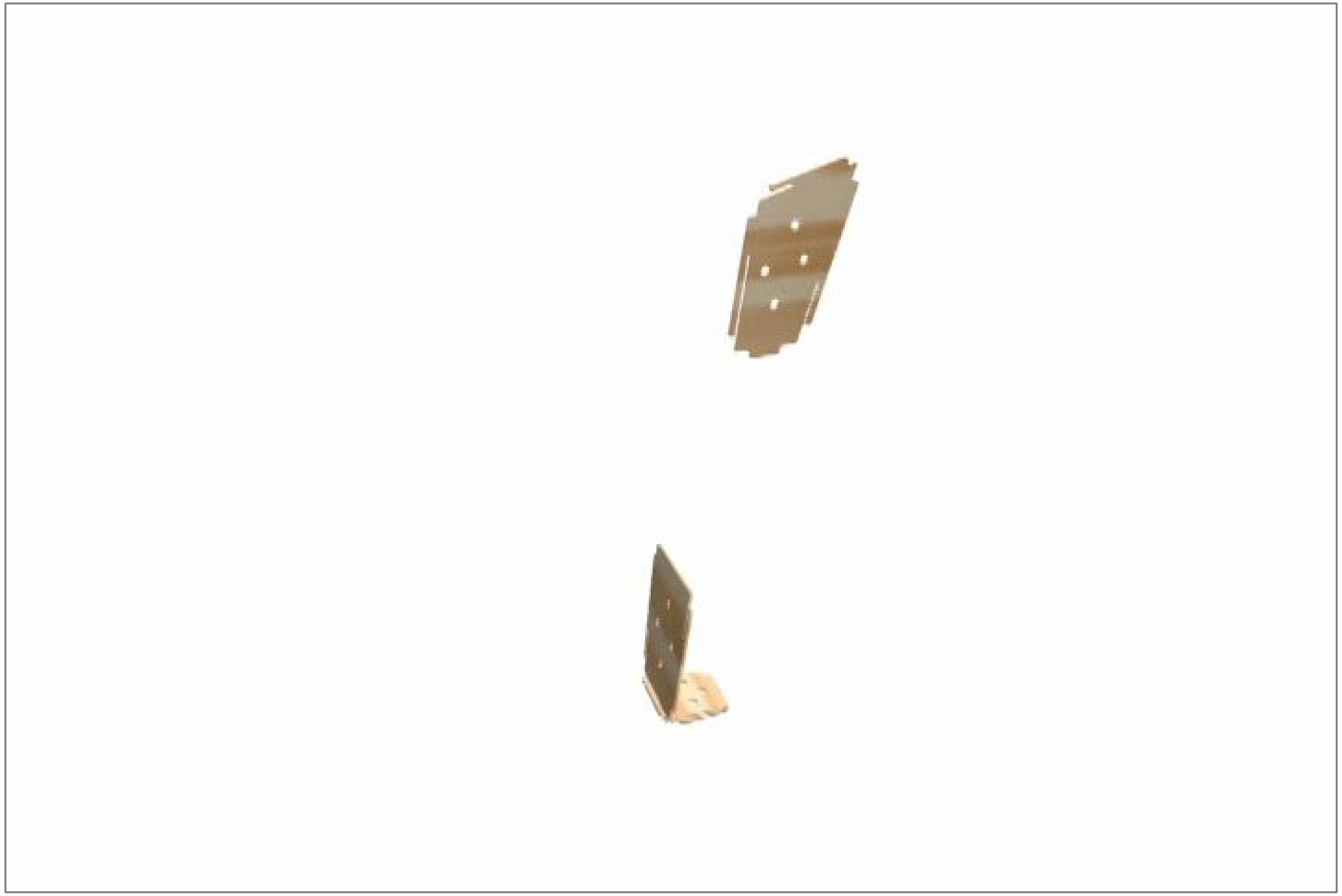
# Computational Interlocking Furniture Assembly

Fu and Song et al.    SIGGRAPH 2015

# CofiFab: Coarse-to-Fine Fabric. of Large 3D Objects

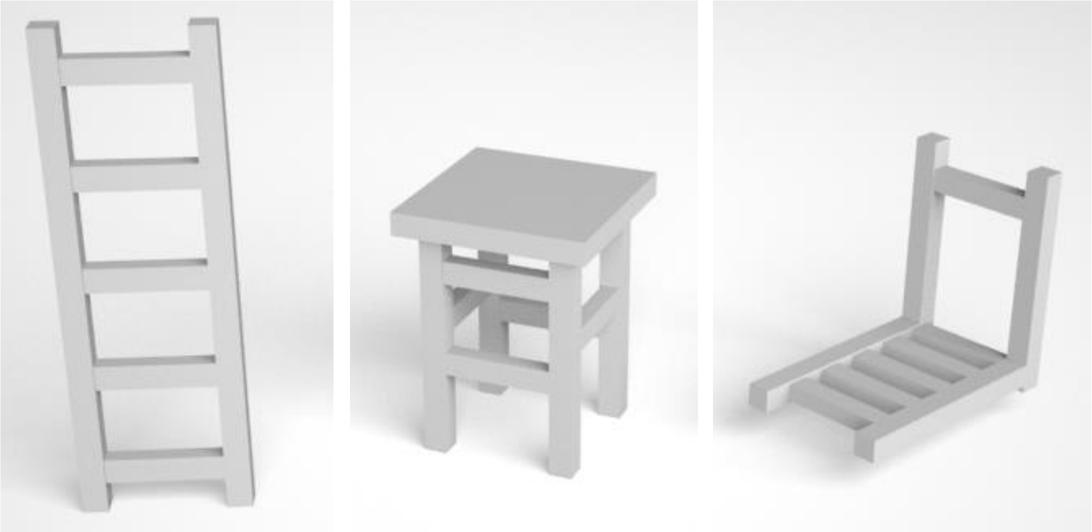
Song et al.    SIGGRAPH 2016



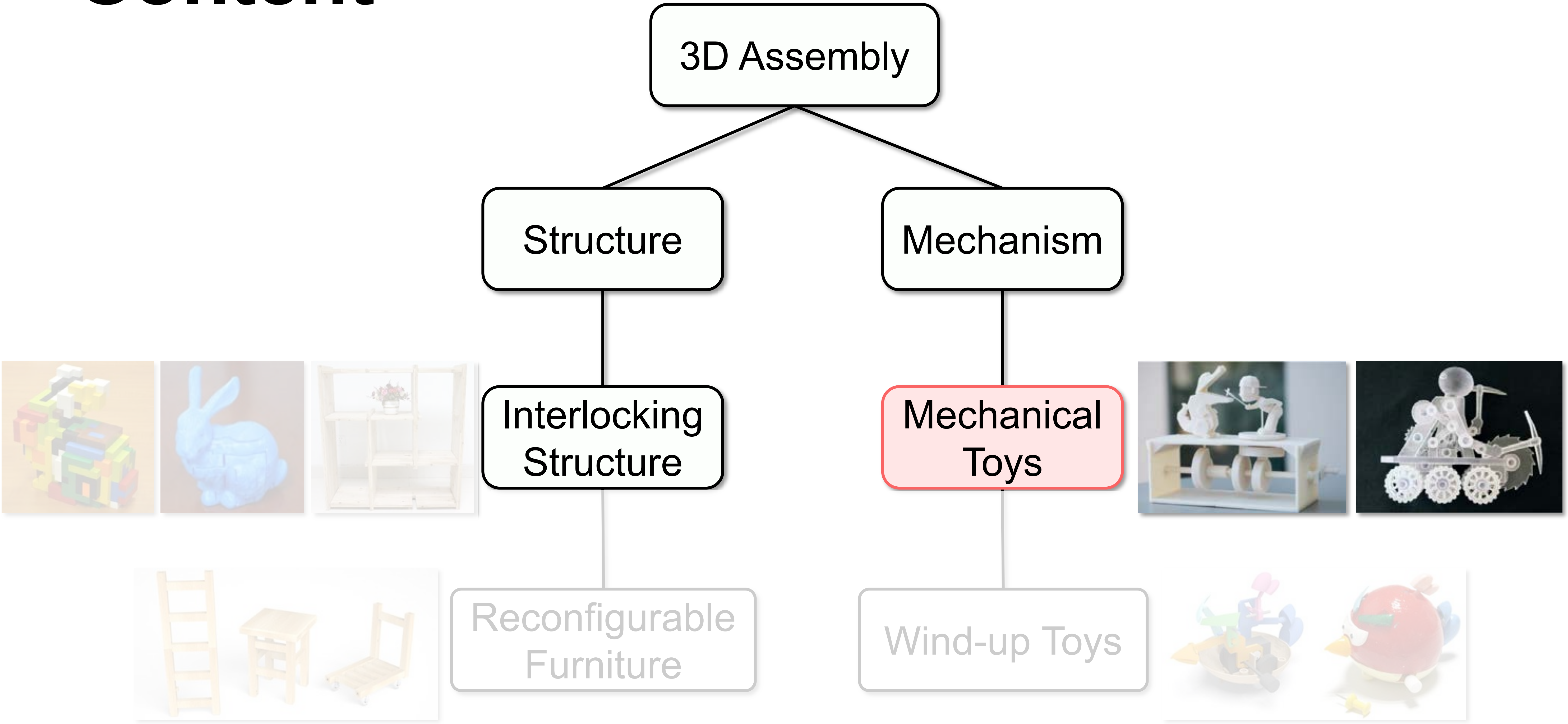


# Reconfigurable Interlocking Furniture

Song et al.    SIGGRAPH Asia 2017



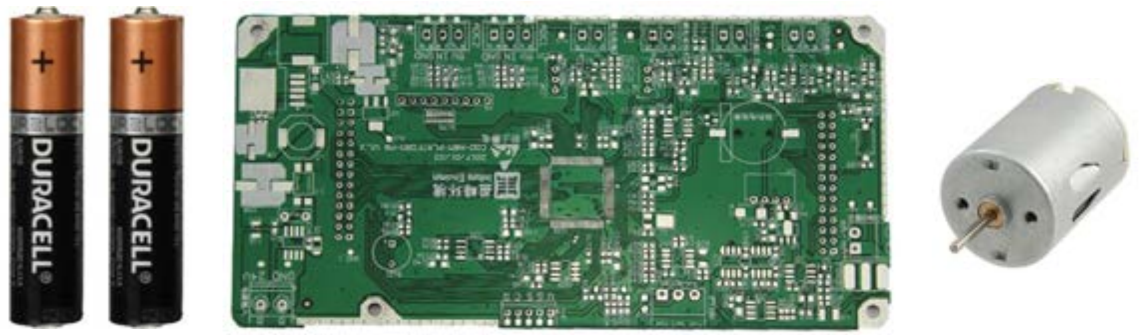
# Content


3D Assembly
Structure
Mechanism
Interlocking Structure
Mechanical Toys
Reconfigurable Furniture
Wind-up Toys

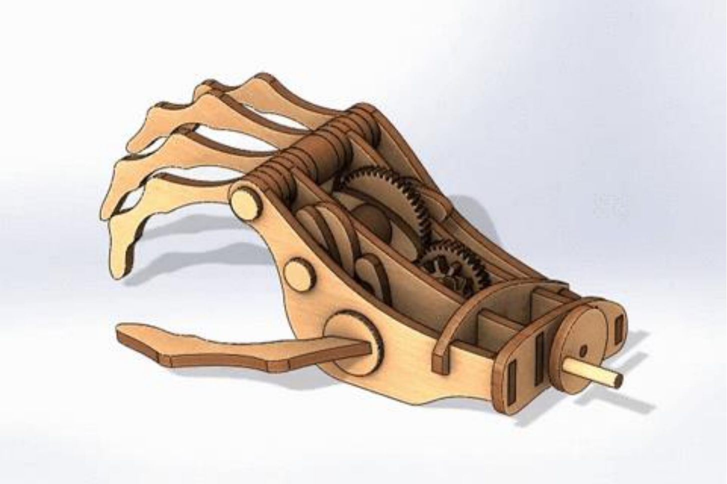
# Mechanical Toy

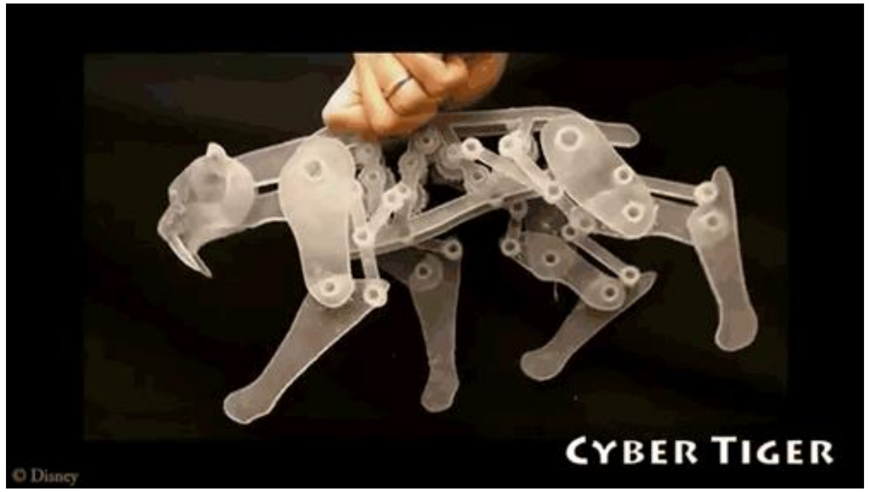
Mechanical toys are toys powered by mechanical energy.

**Electric** toy VS **Mechanical** toy



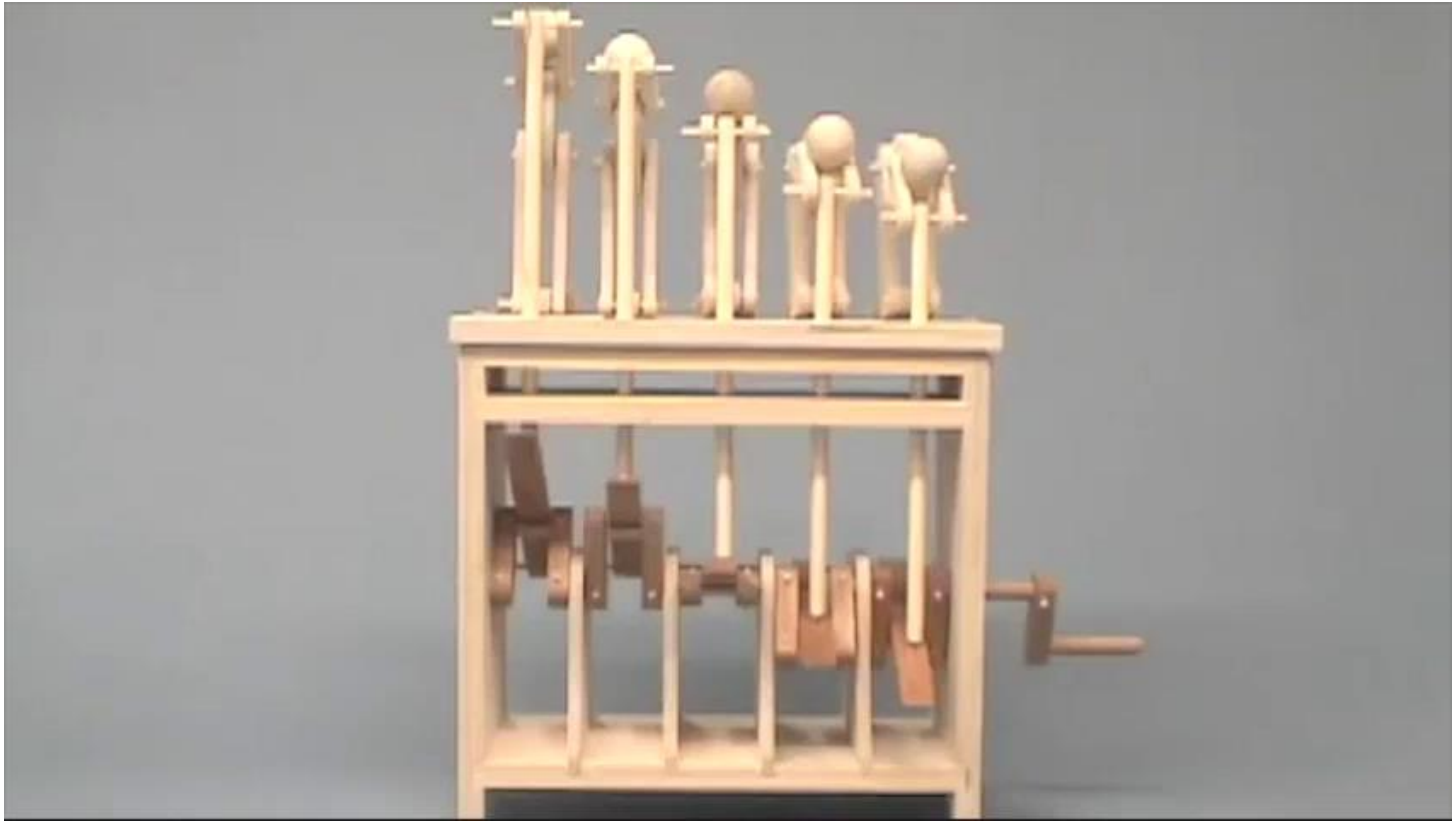

# Mechanical Toy



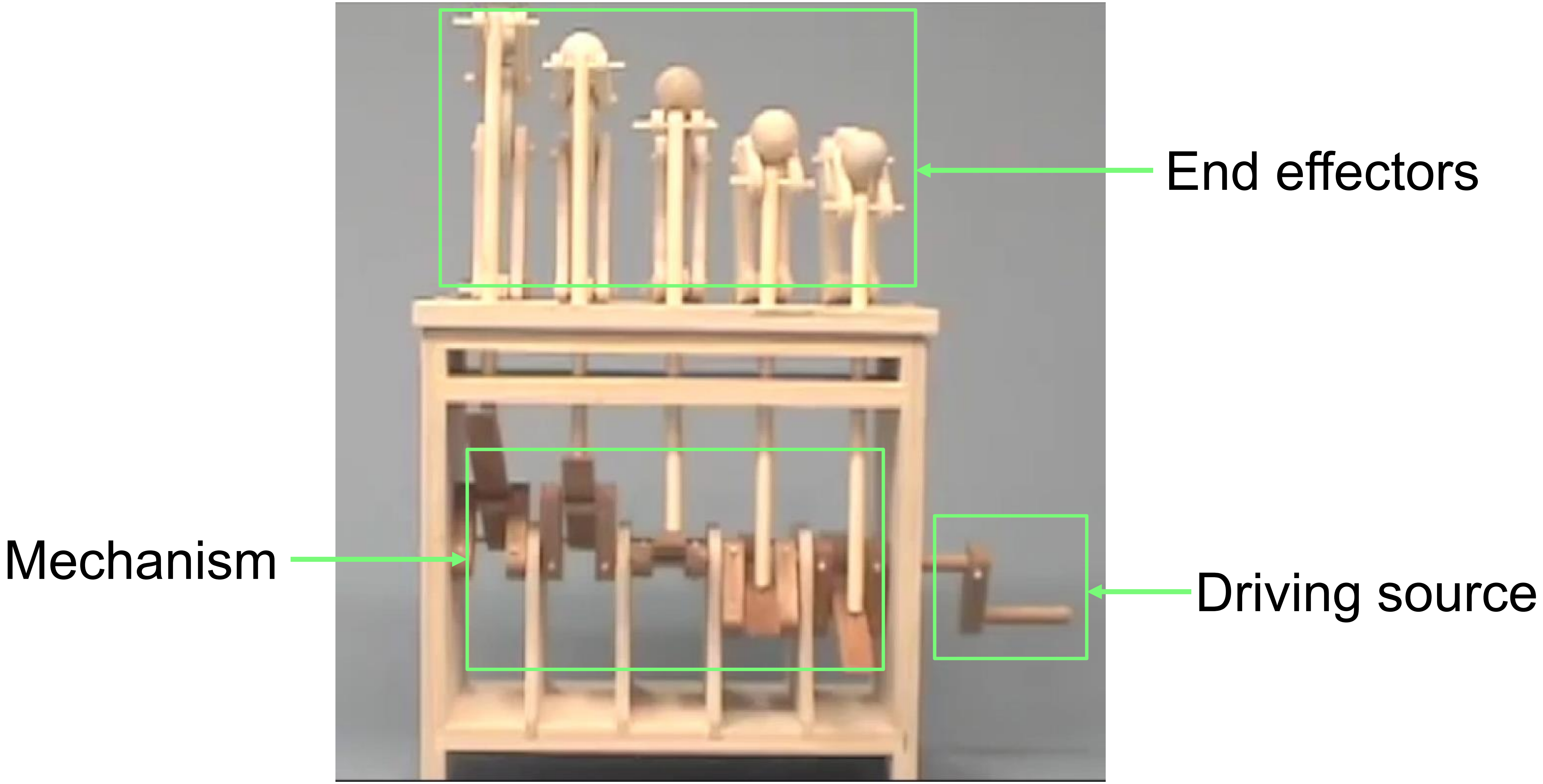




# Example Toy

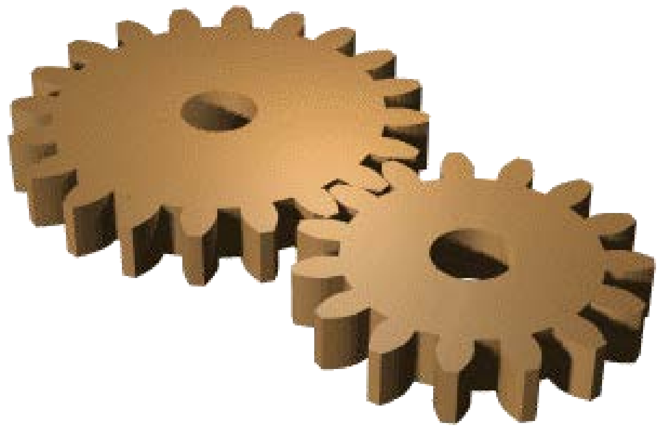


# Example Toy

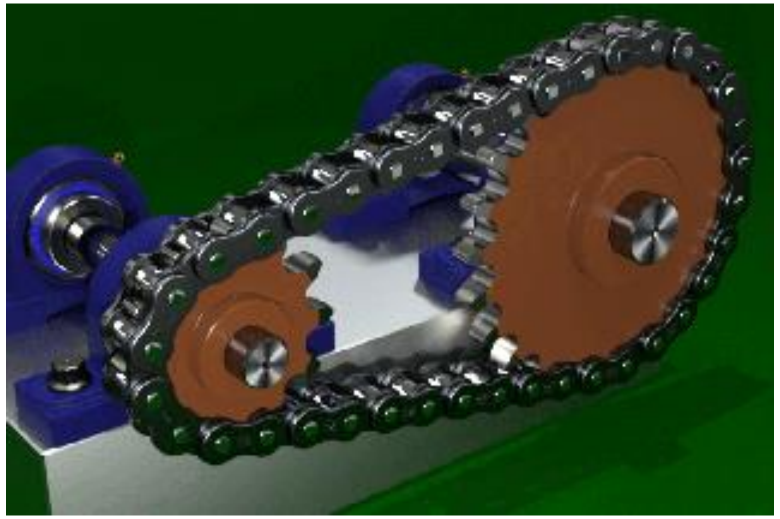


# Typical Elemental Mechanisms

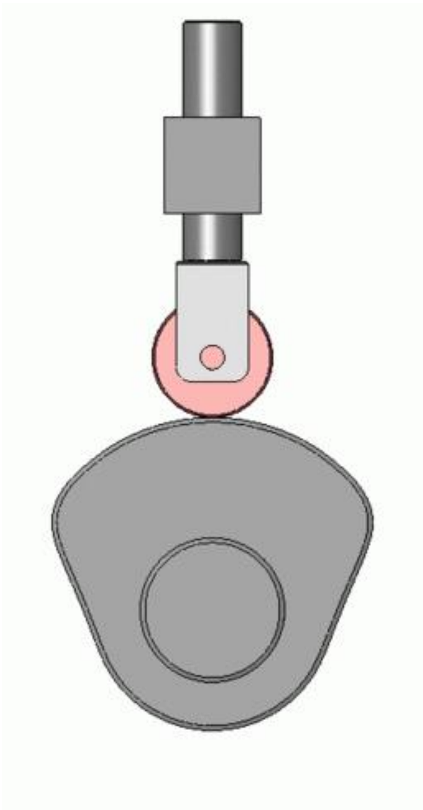
An elemental mechanism delivers a fundamental motion transfer task with two or more mechanical parts



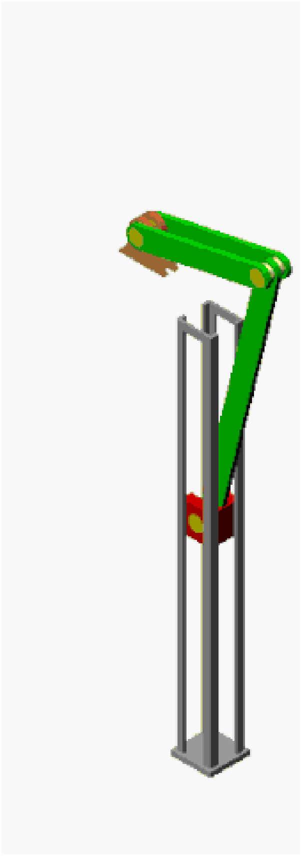
Gears



Belt and pulley



Cam



Linkage

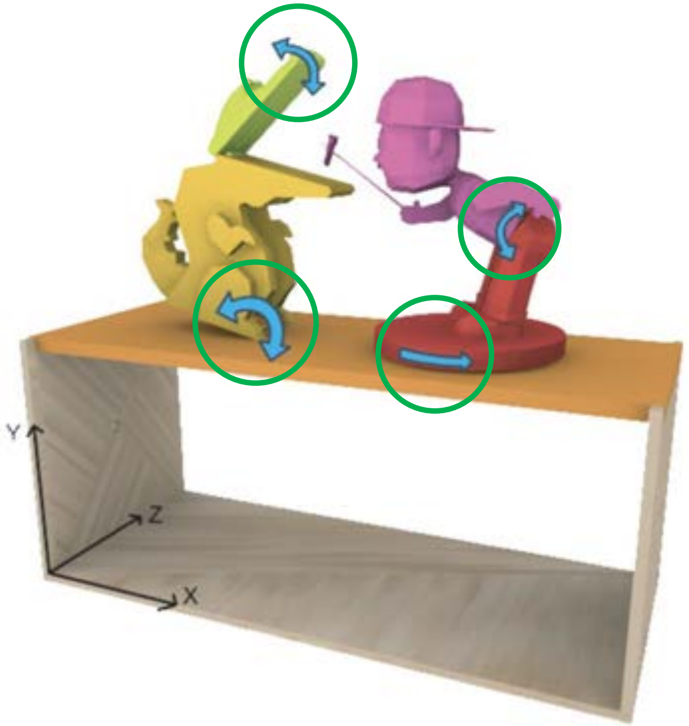
# Connect Elemental Mechanisms

A more complex mechanism can be built by connecting a number of elemental mechanisms.

# **Problem**: Mechanism Design

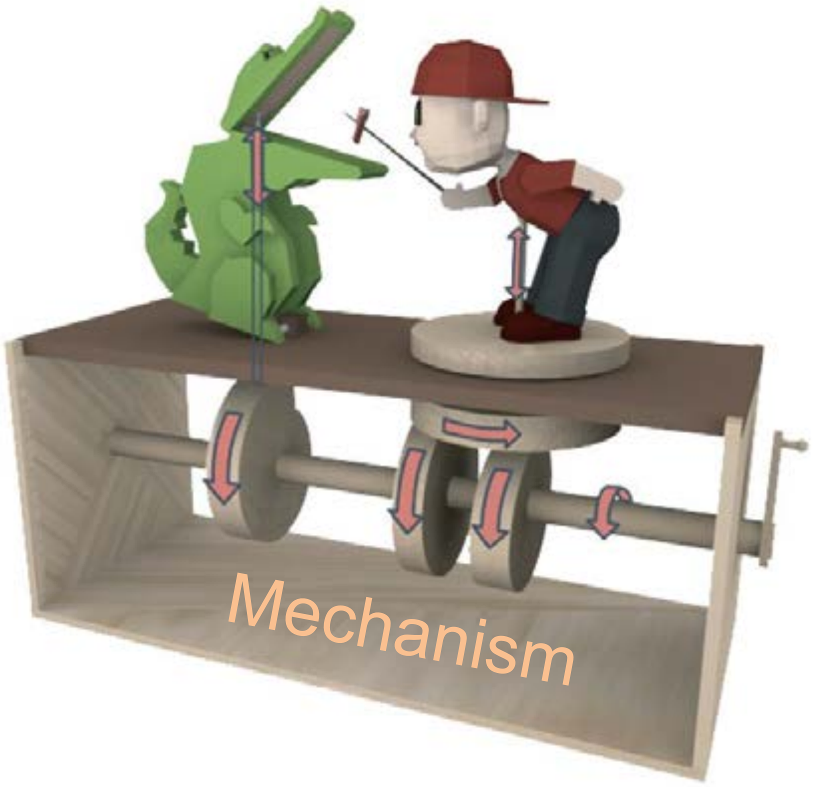
**Input**:

1. Toy **shape**
2. End-effector **motions**
3. Library of elem. Mechanism




**Output**:

**Mechanism** that transfers motion from driving source to end-effectors




[Zhu et al. 2012]

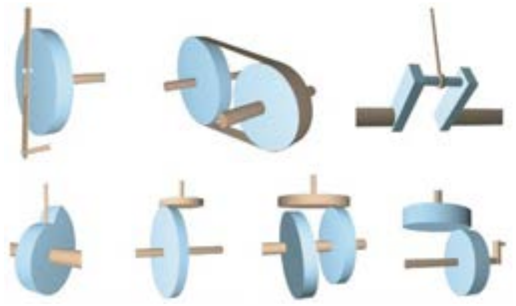
# General Approach

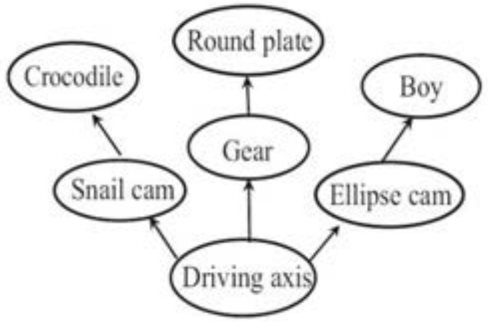
Model relation between geometry and kinematics

Select and connect elemental mechanisms

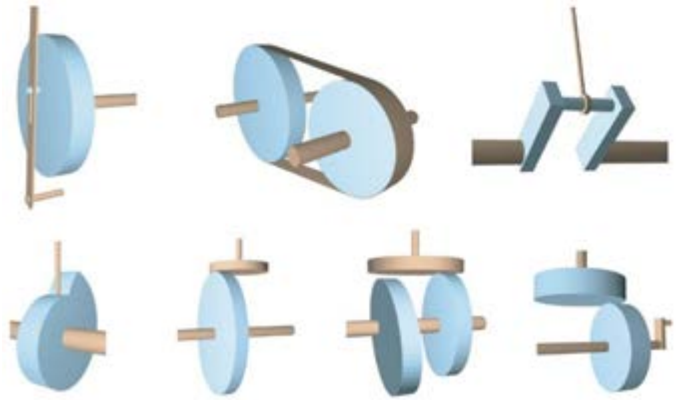
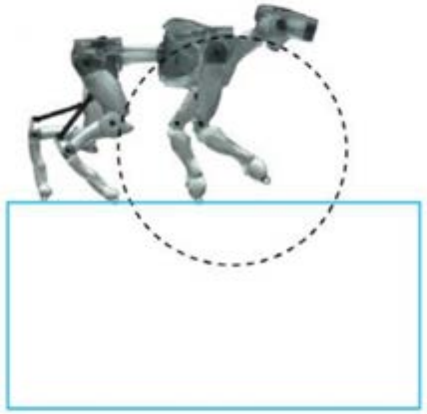
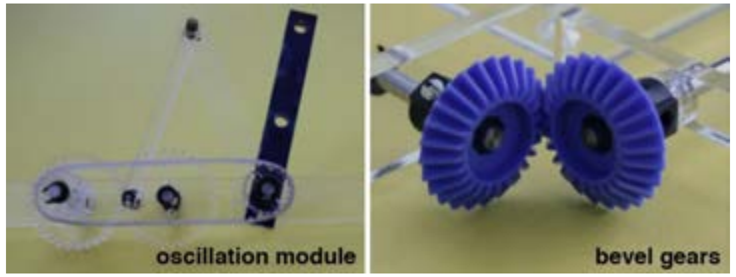
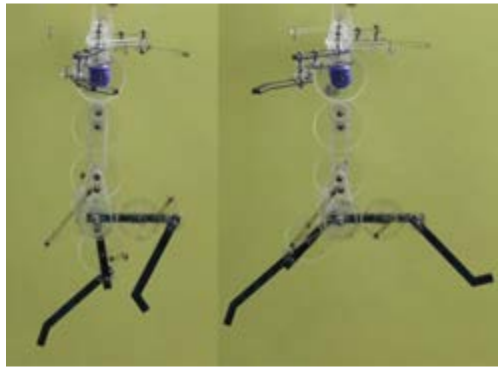
Optimize parameters of elemental mechanisms

Input → Model elemental mechanisms → Conceptual design → Dimensional synthesis → Output

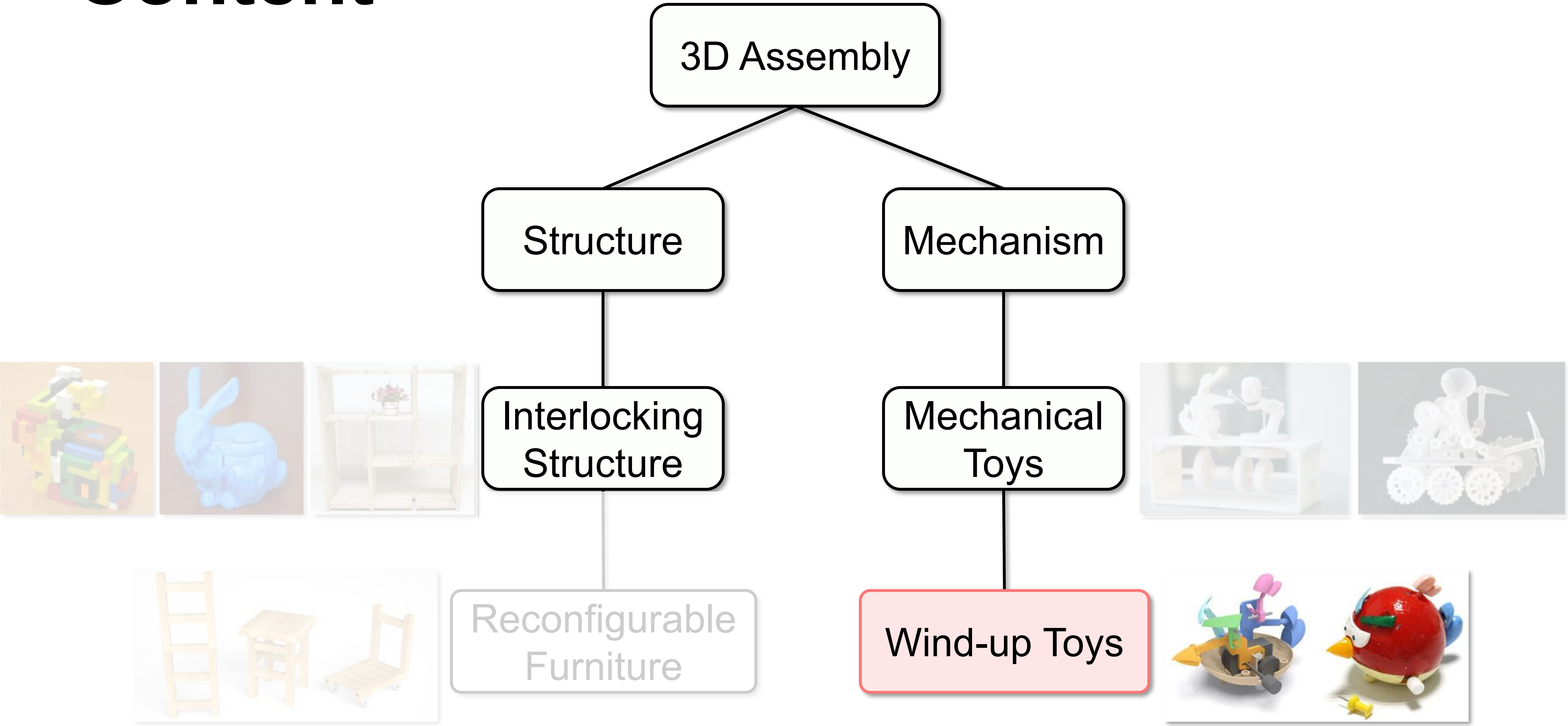





# Related Works

| | | Elemental mechanisms | Results |
|---|---|---|---|
| [Zhu et al. 2012] |  |  |  |
| [Coros et al. 2013] |  |  |  |
| [Ceylan et al. 2013] |  |   |  |

# Content

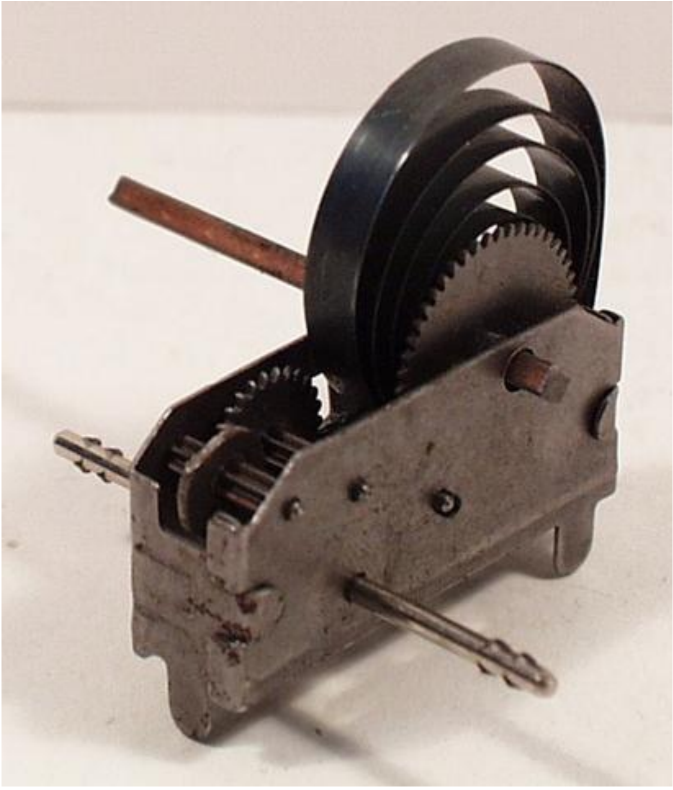

3D Assembly
Structure
Mechanism
Interlocking Structure
Mechanical Toys
Reconfigurable Furniture
Wind-up Toys

# Wind-up Toys

# Wind-up Toys

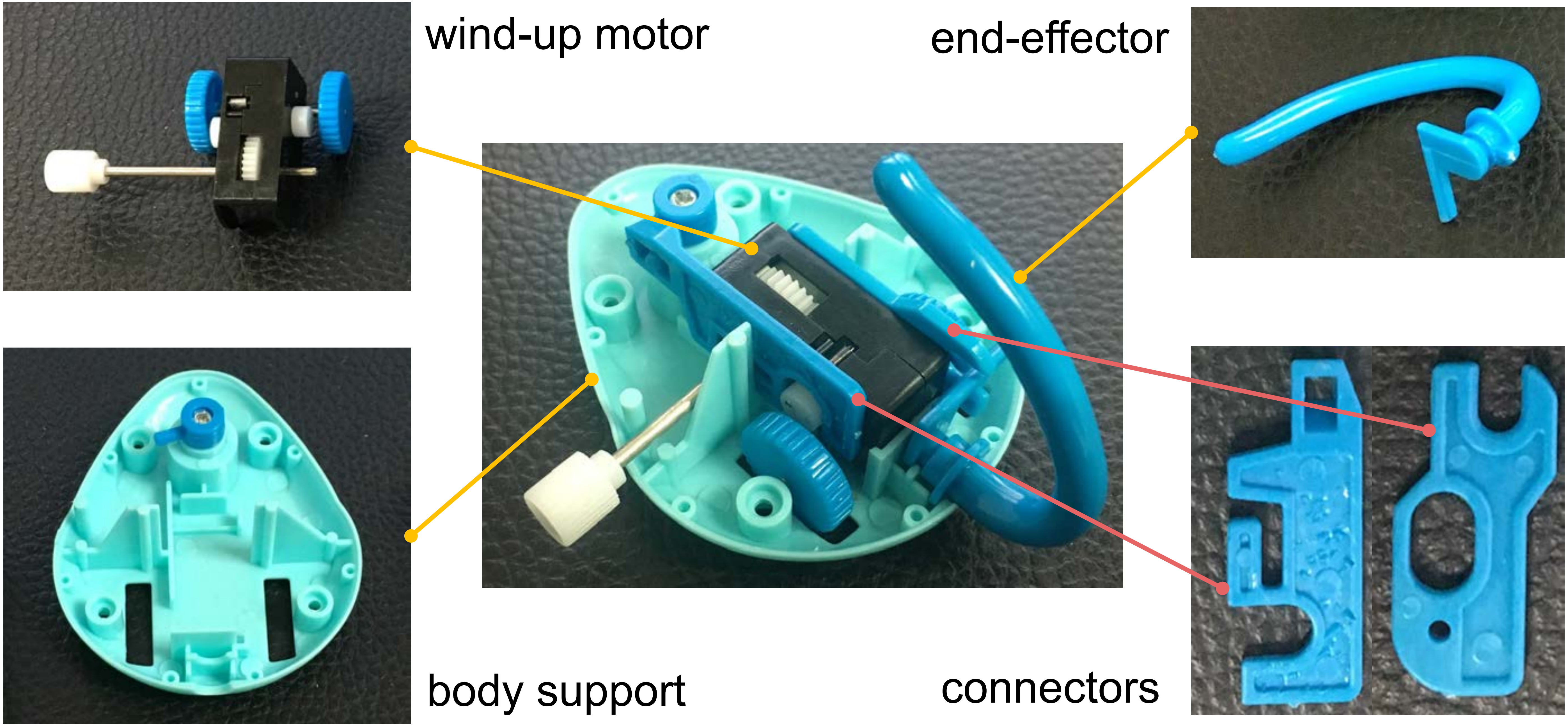
Powered by a clockwork motor

clockwork motor



# Wind-up Toys

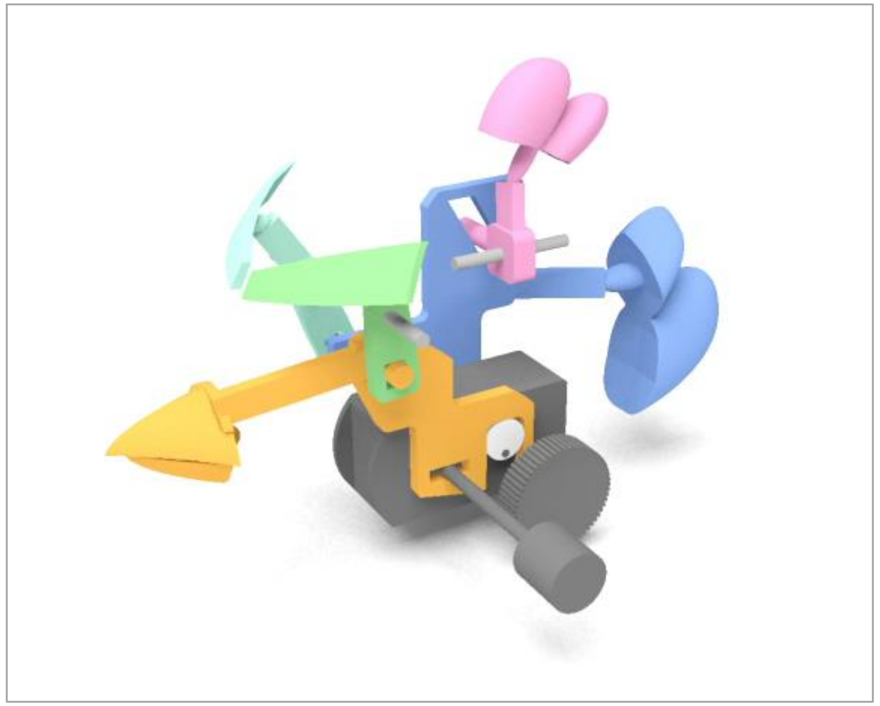
# How it Works?



# Our Goal

Develop a computational tool to aid the design and fabrication of wind-up toys, focusing on automated construction of the internal mechanisms

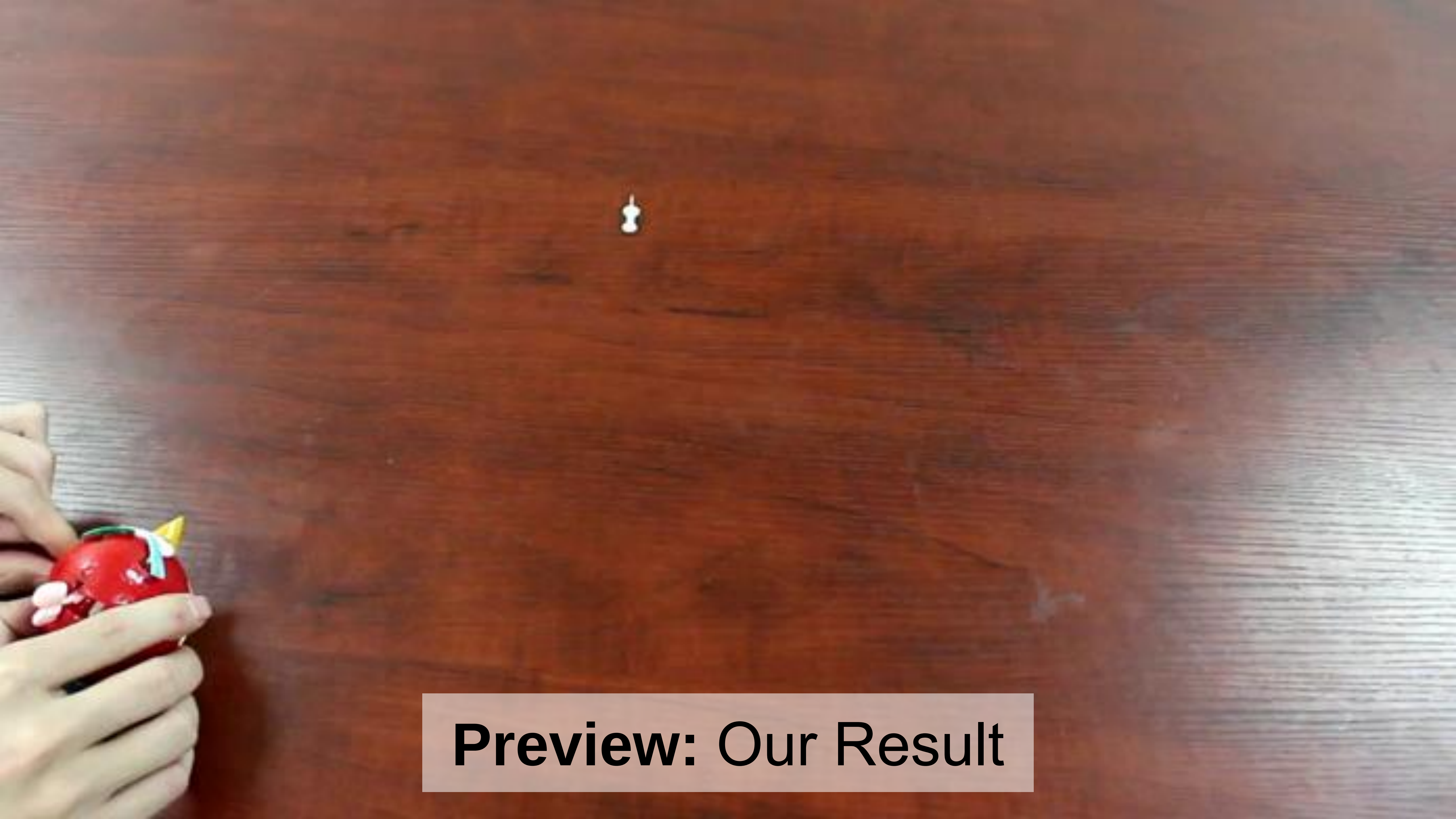
User inputs



Mechanism design

Fabrication

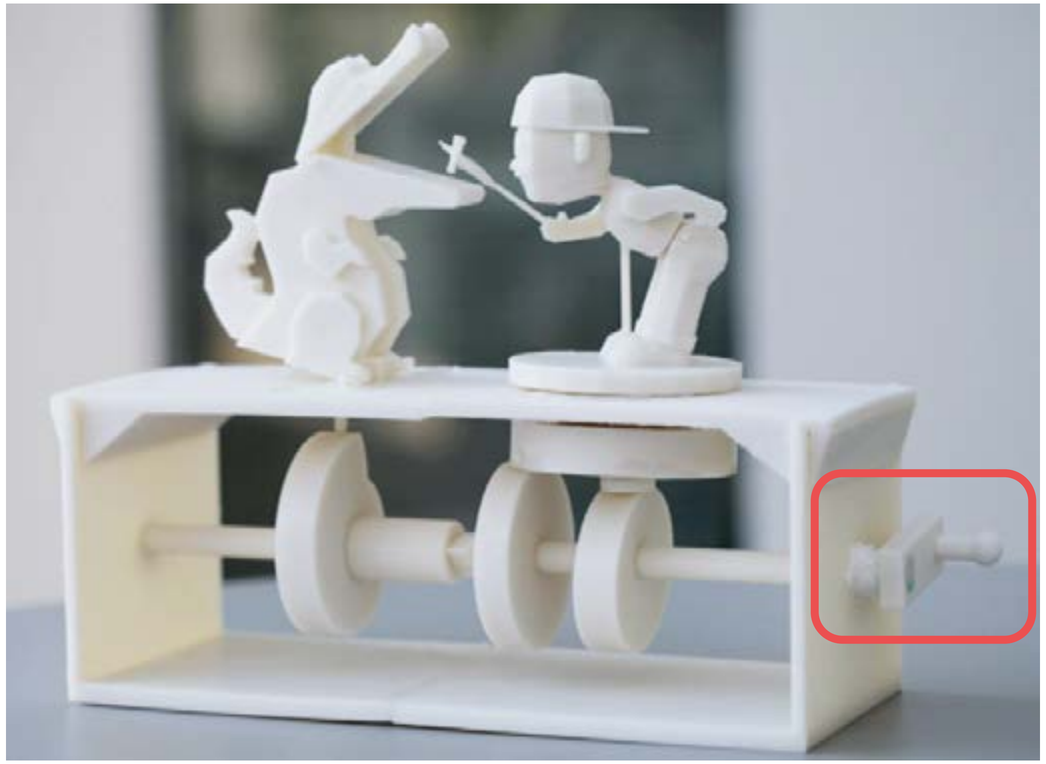
# **Preview:** Our Result


**Preview:** Our Result

# **Challenge #1:** Small Driving Force

Wind-up motor can only store limited energy and provide a small torque,

Hand-operated crank



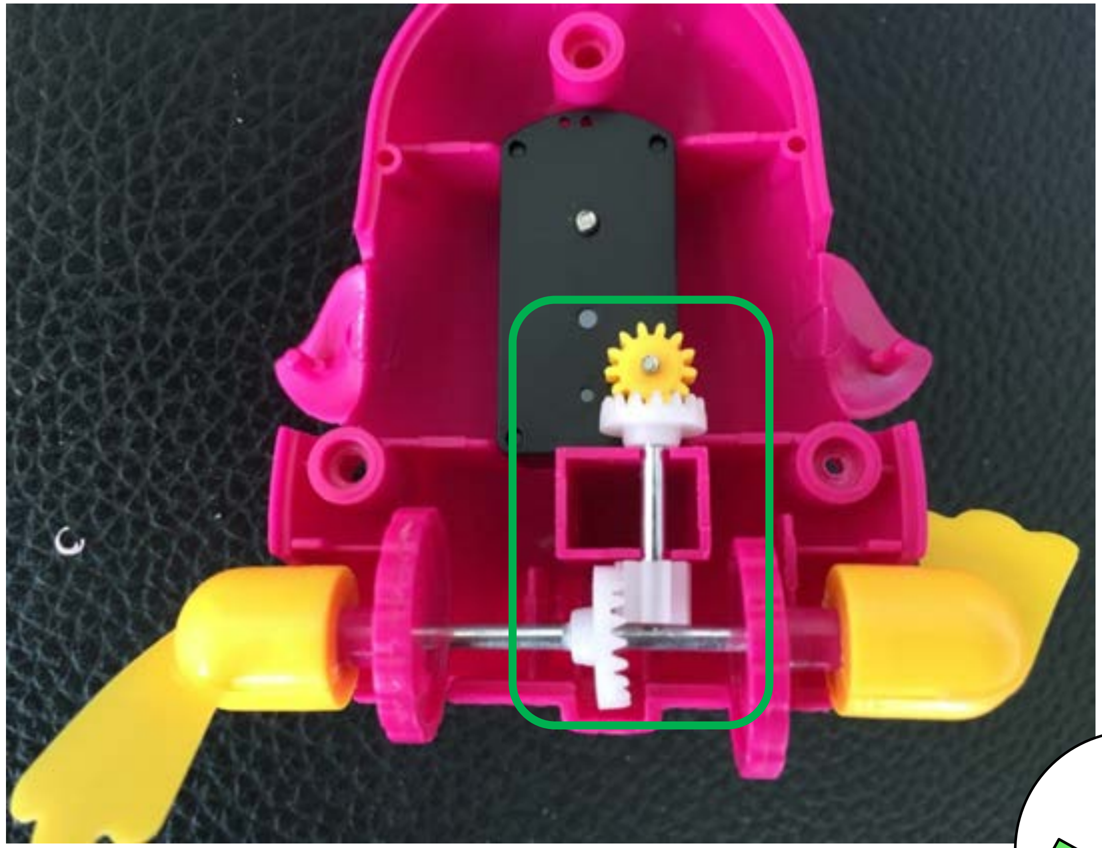
[Zhu et al. 2012]

Electric drill

[Coros et al. 2013]

# **Challenge #1:** Small Driving Force

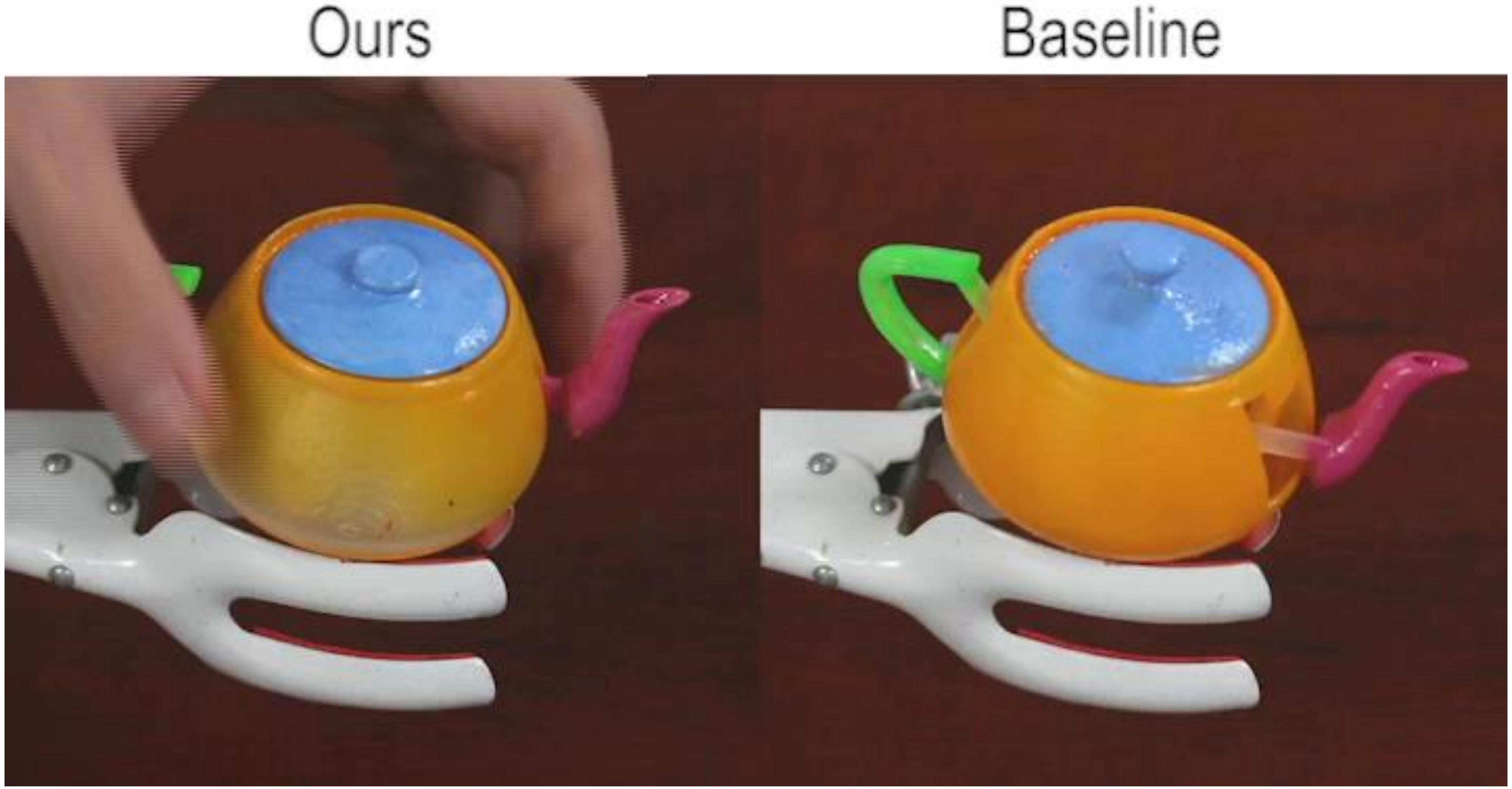
Wind-up motor can only store limited energy and provide a small torque, thus requiring a lightweight wind-up mechanism.



[Coros et al. 2013]

# **Challenge #1:** Small Driving Force

By this, the wind-up toy can perform motions for a longer duration.

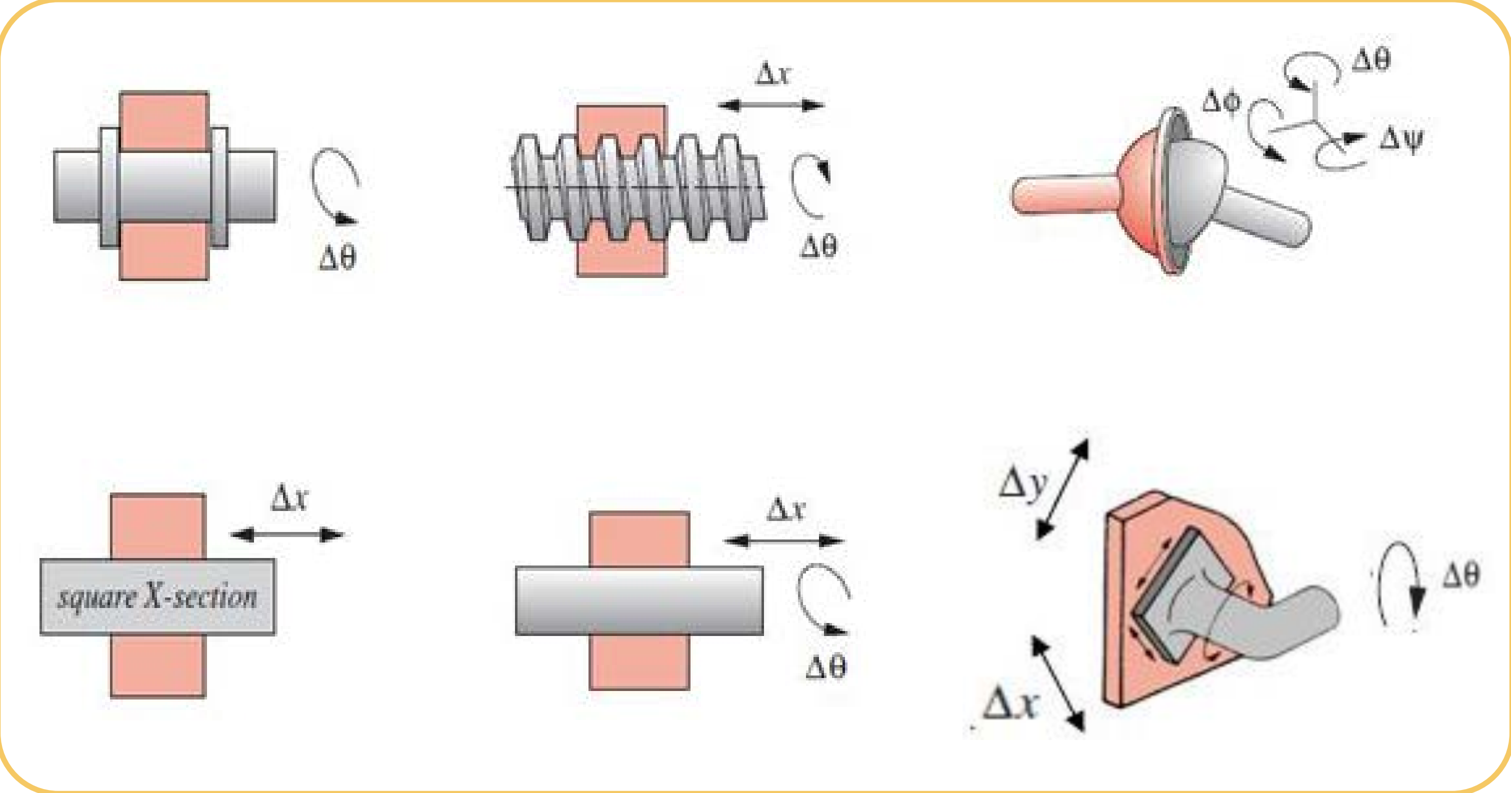


# **Challenge #2:** Higher Pair Joints

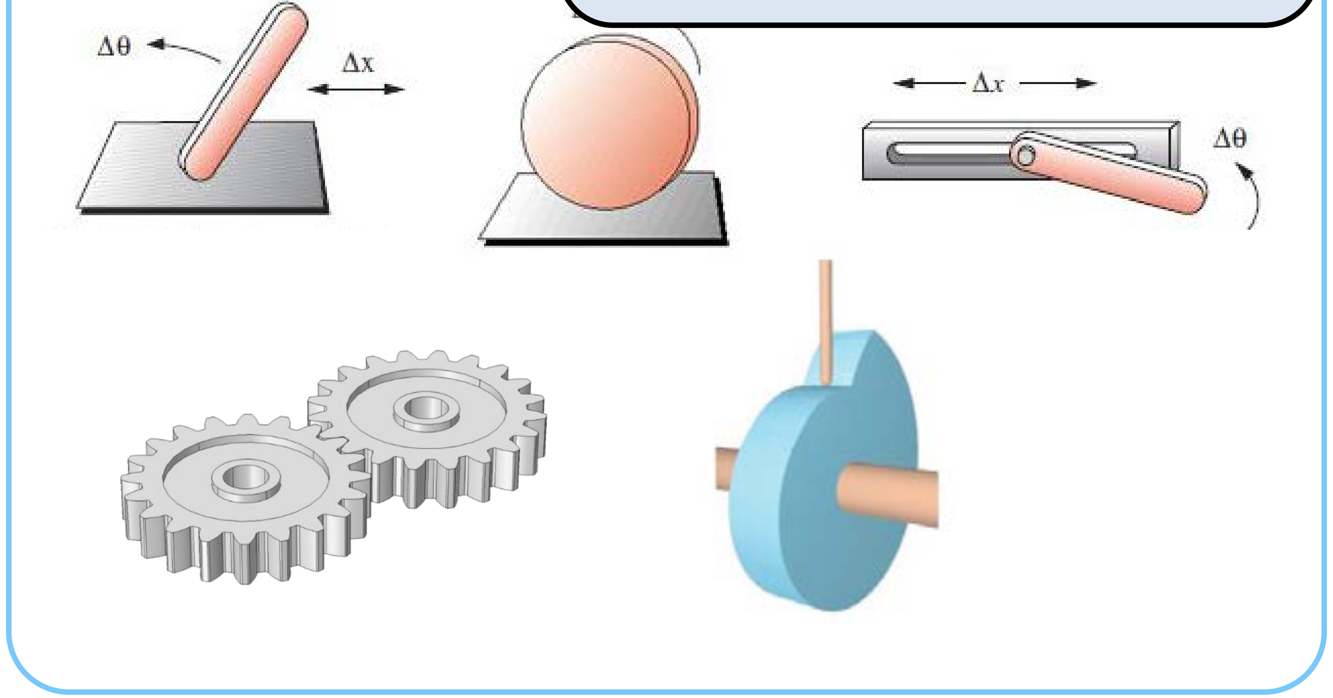
Wind-up toys heavily employ higher pair joints to transfer motion;

Higher pair joints: point or line contact between associ

Less friction
Higher Flexibility
Smaller driving force




Lower pairs

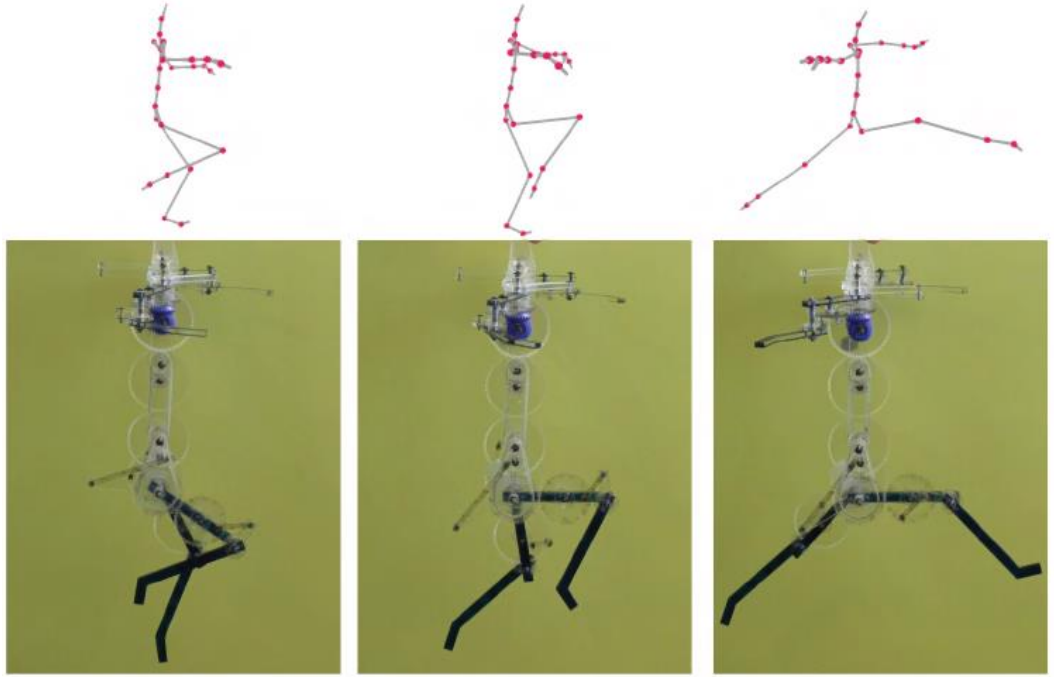



Higher pairs

# **Challenge #2:** Higher Pair Joints

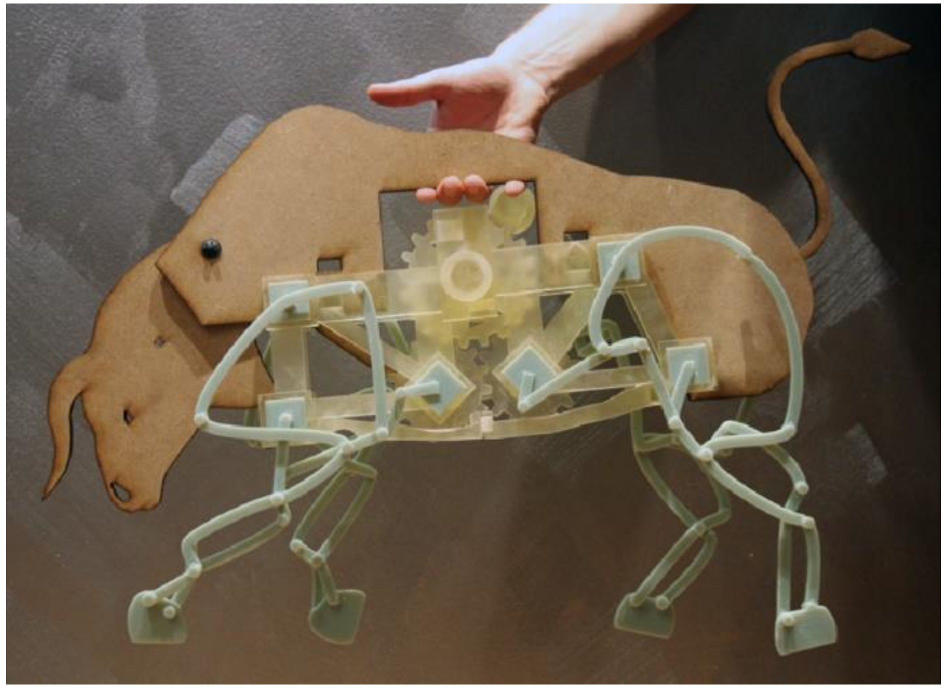
Wind-up toys heavily employ higher pair joints to transfer motion, which has not been fully explored in previous works.

Pulleys and gears



[Ceylan et al. 2013]

Linkages



[Thomaszewski et al. 2014]

# **Challenge #3:** Compact Mechanism

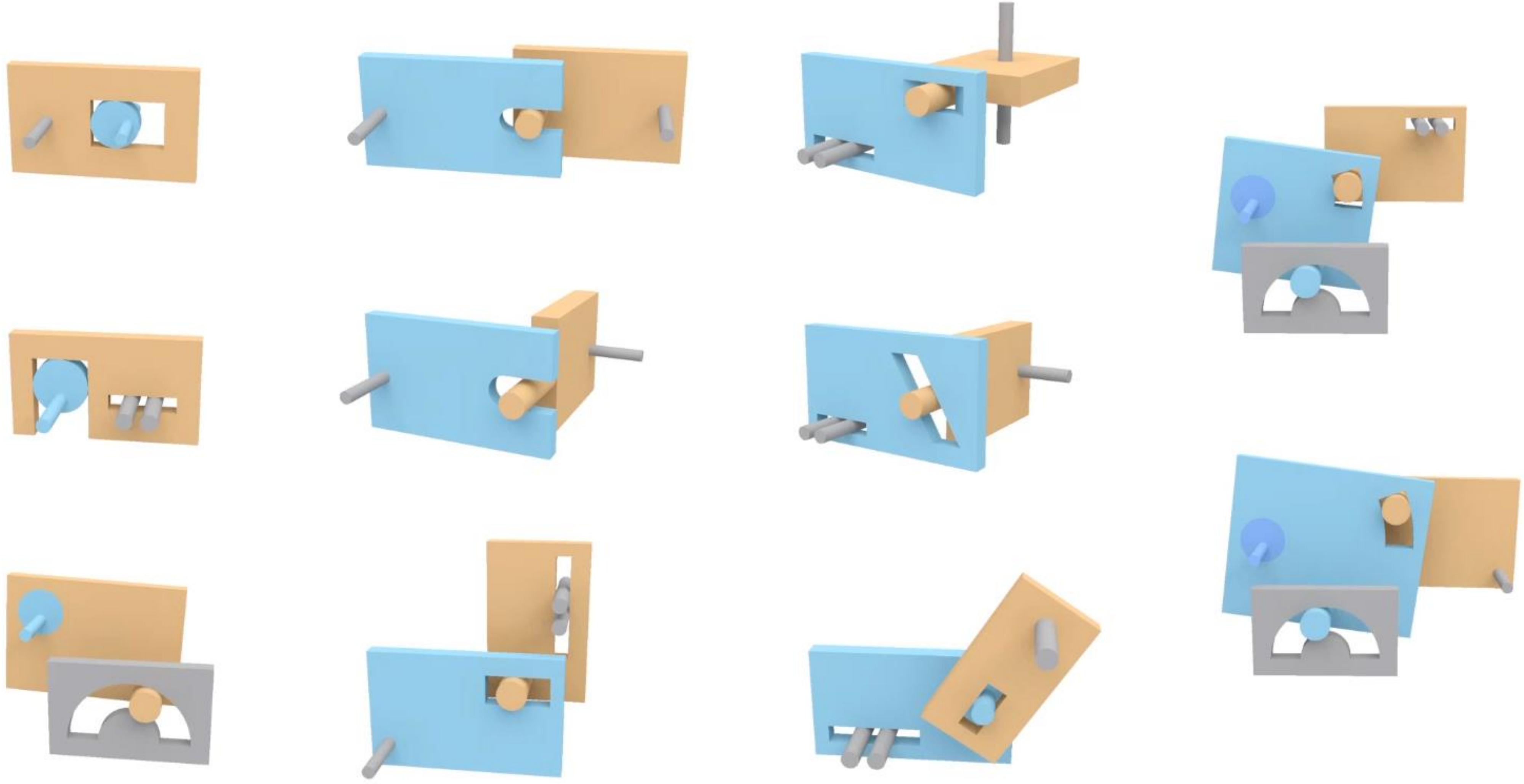
Wind-up mechanism should be enclosed inside the small toy body,

# **Challenge #3:** Compact Mechanism

Wind-up mechanism should be enclosed inside the small toy body, thus calling for a compact wind-up mechanism

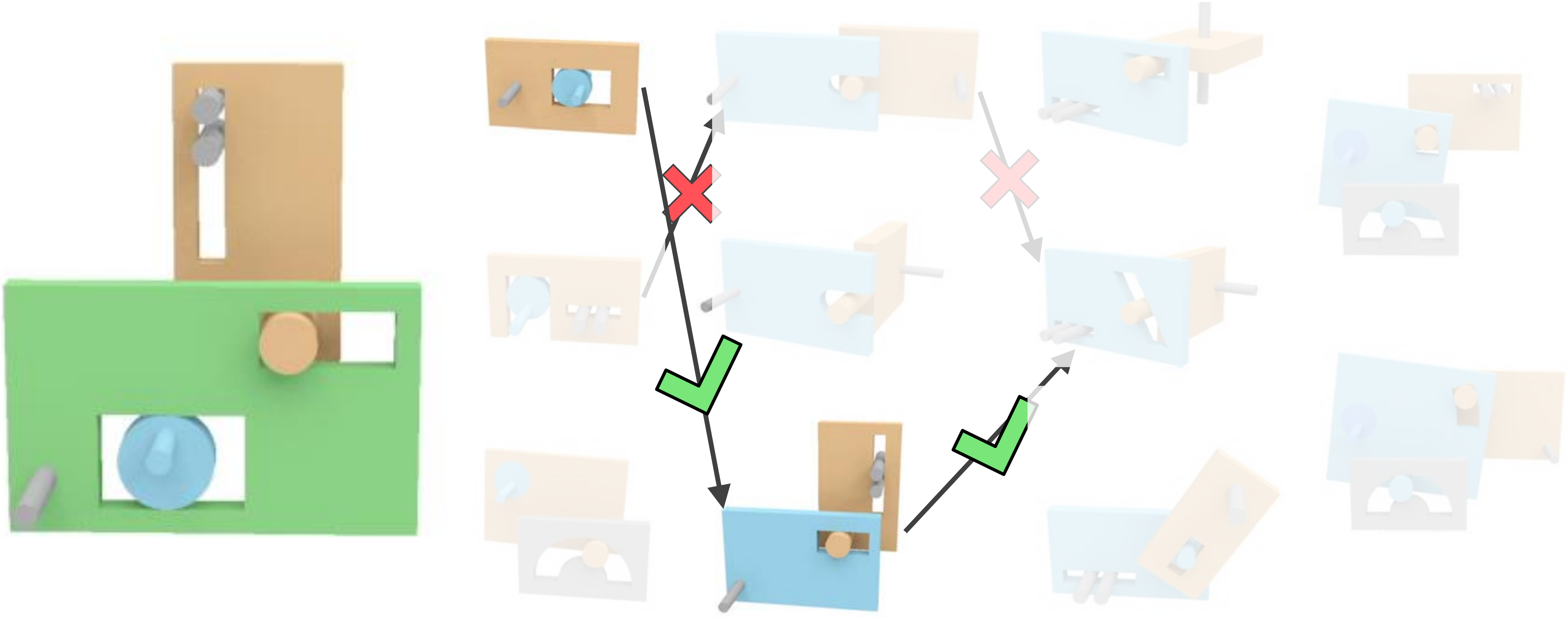
# **Key Idea #1:** Model Elemental Mechanisms

Model elemental mechanisms with higher pair joints as building blocks



# **Key Idea #2:** Connect Elemental Mechanisms

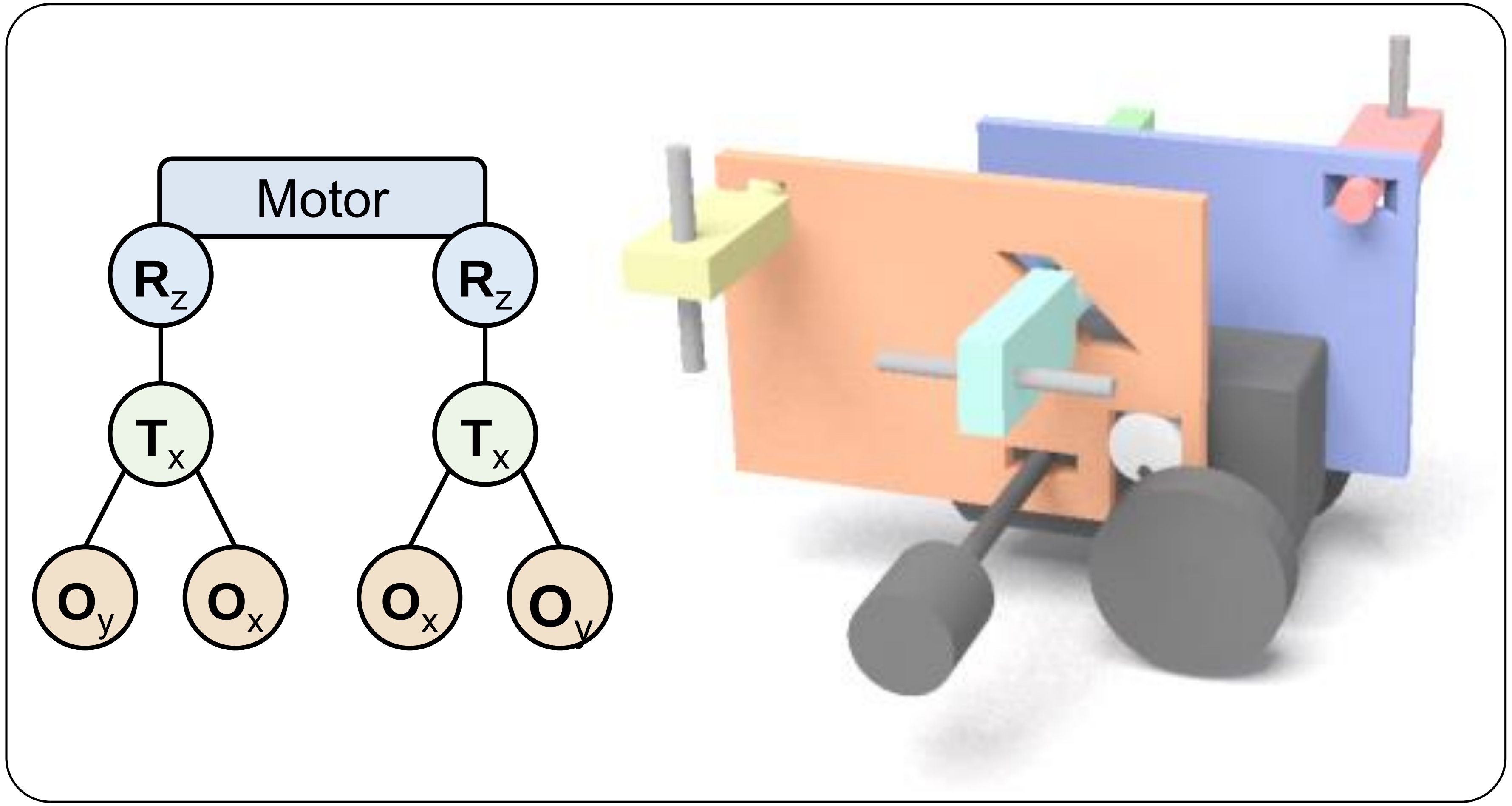
Construct a wind-up mechanism by connecting multiple elem. mechanisms



# **Key Idea #3:** Optimize Wind-up Mechanisms

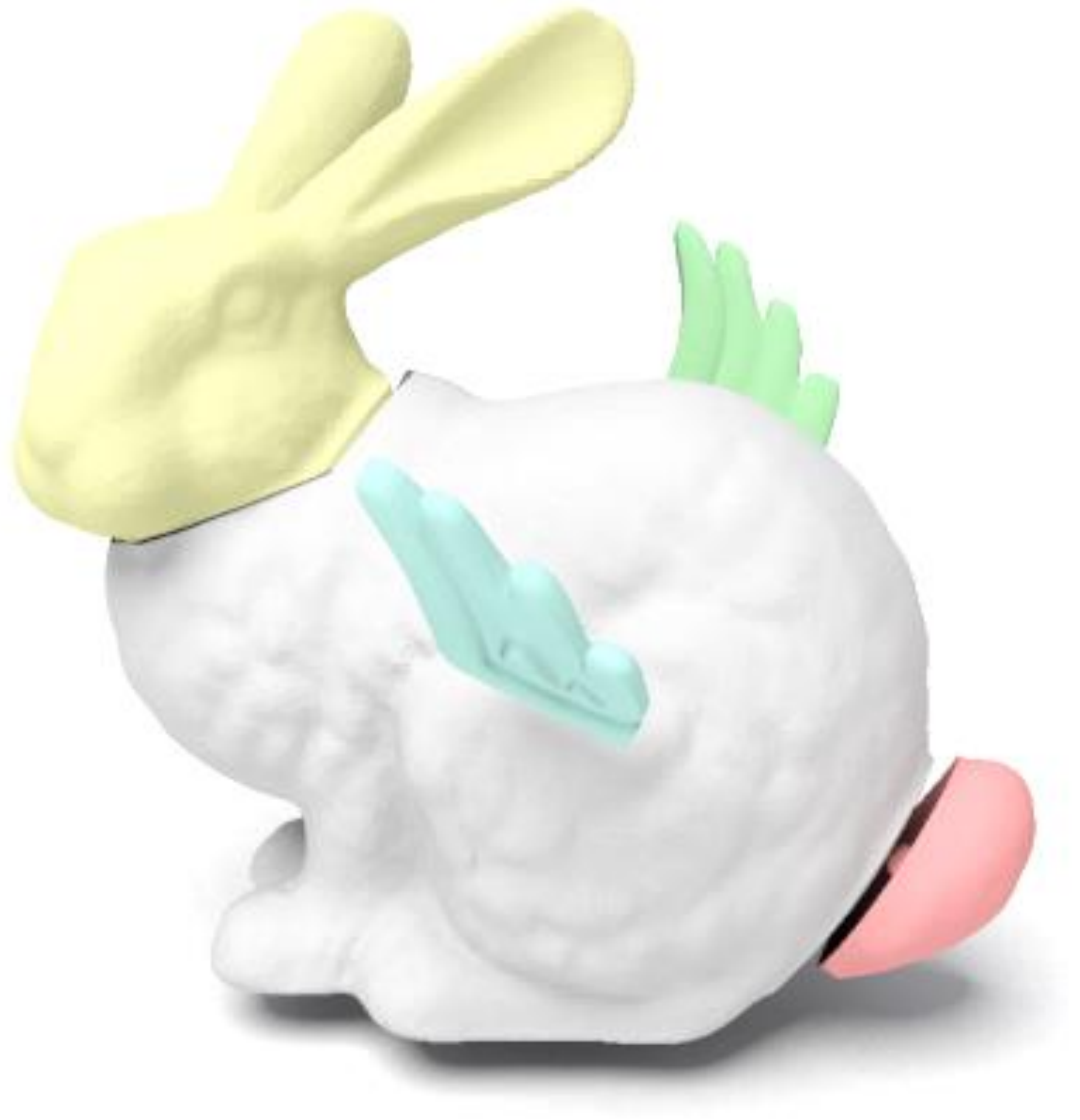
Optimize topology and geometry of wind-up mechanisms

Optimize topology



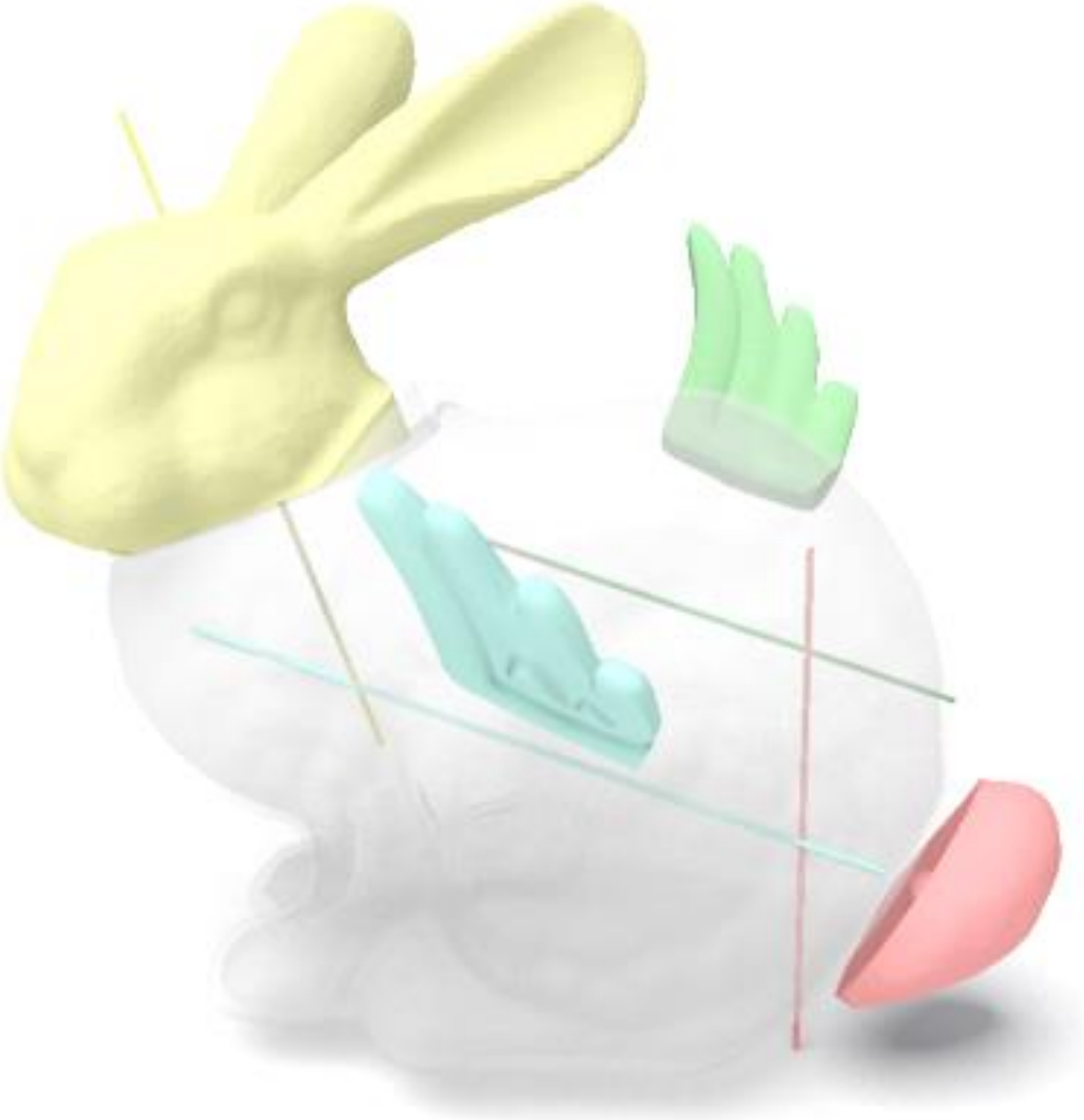

Optimize geometry

# Overview



**Input #1**
Toy parts

# Overview

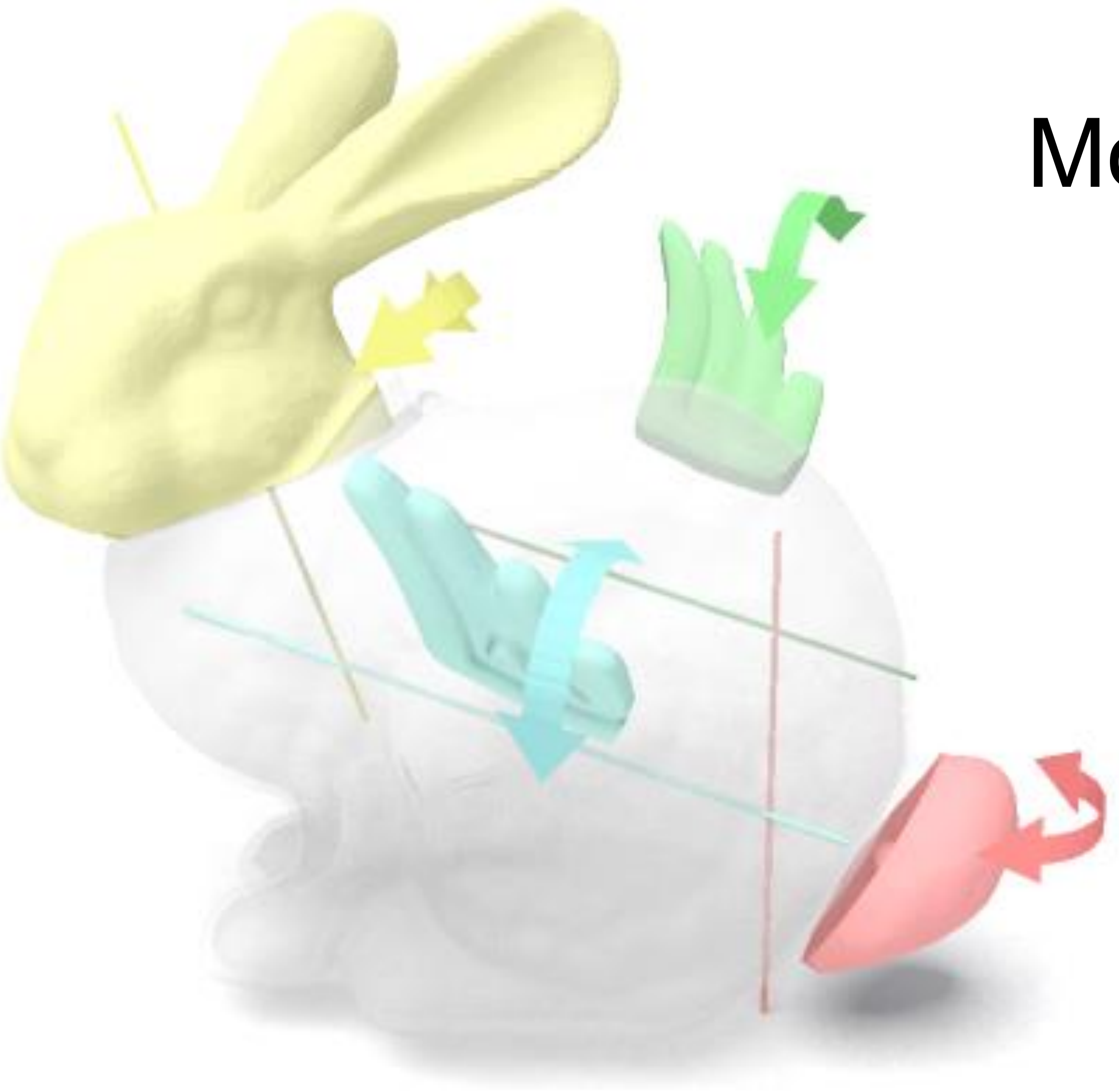


**Input #2**

Motion axes

| Part motion | Symbol |
|---|---|
| rotation (**R**) | |
| oscillating (**O**) | |
| translation (**T**) | |
| oscillating & translation (**OT**) | |

# Overview

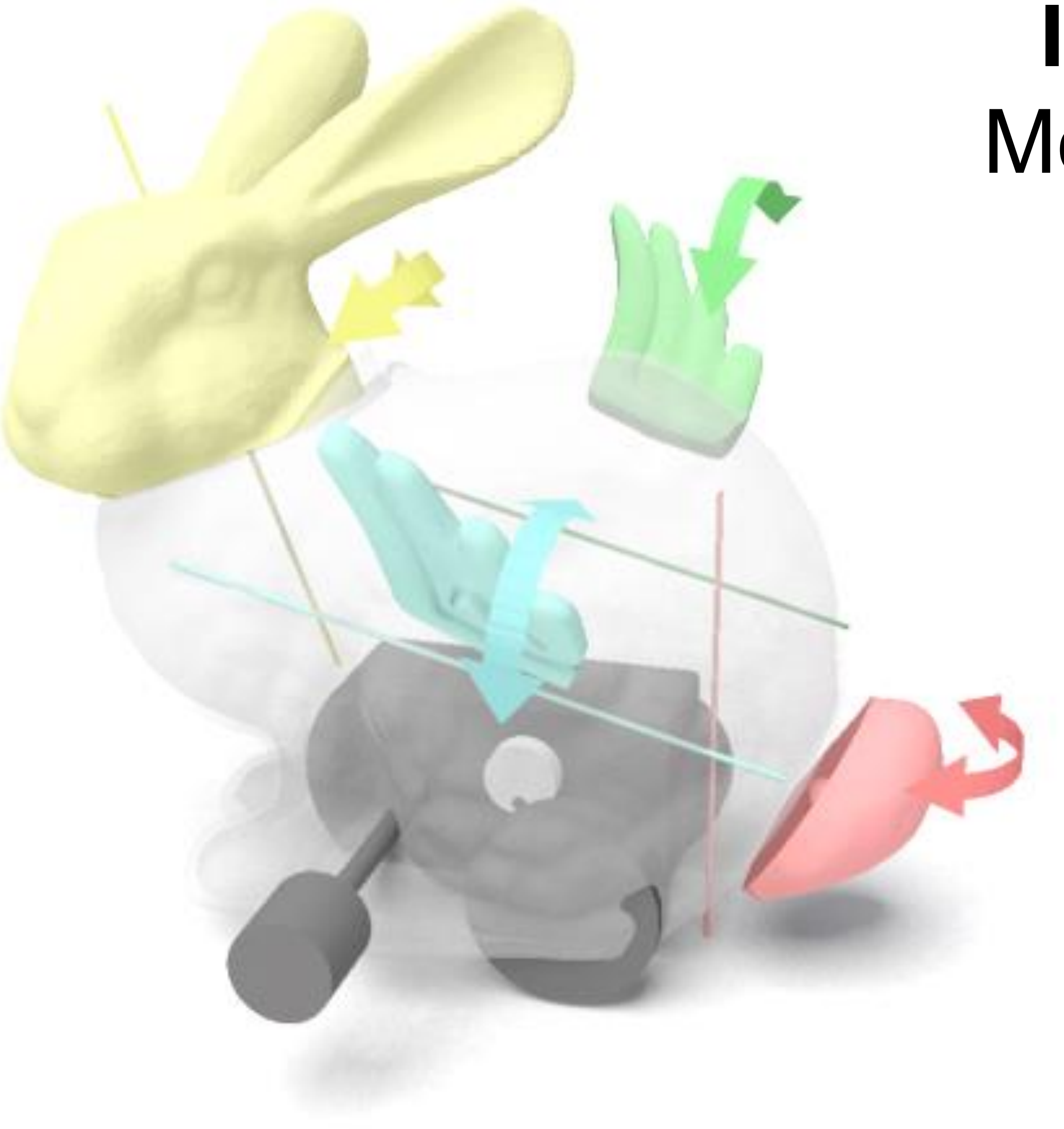


**Input #2**

Motion range

| Part motion | Symbol |
|---|---|
| rotation (**R**) | |
| oscillating (**O**) | |
| translation (**T**) | |
| oscillating & translation (**OT**) | |

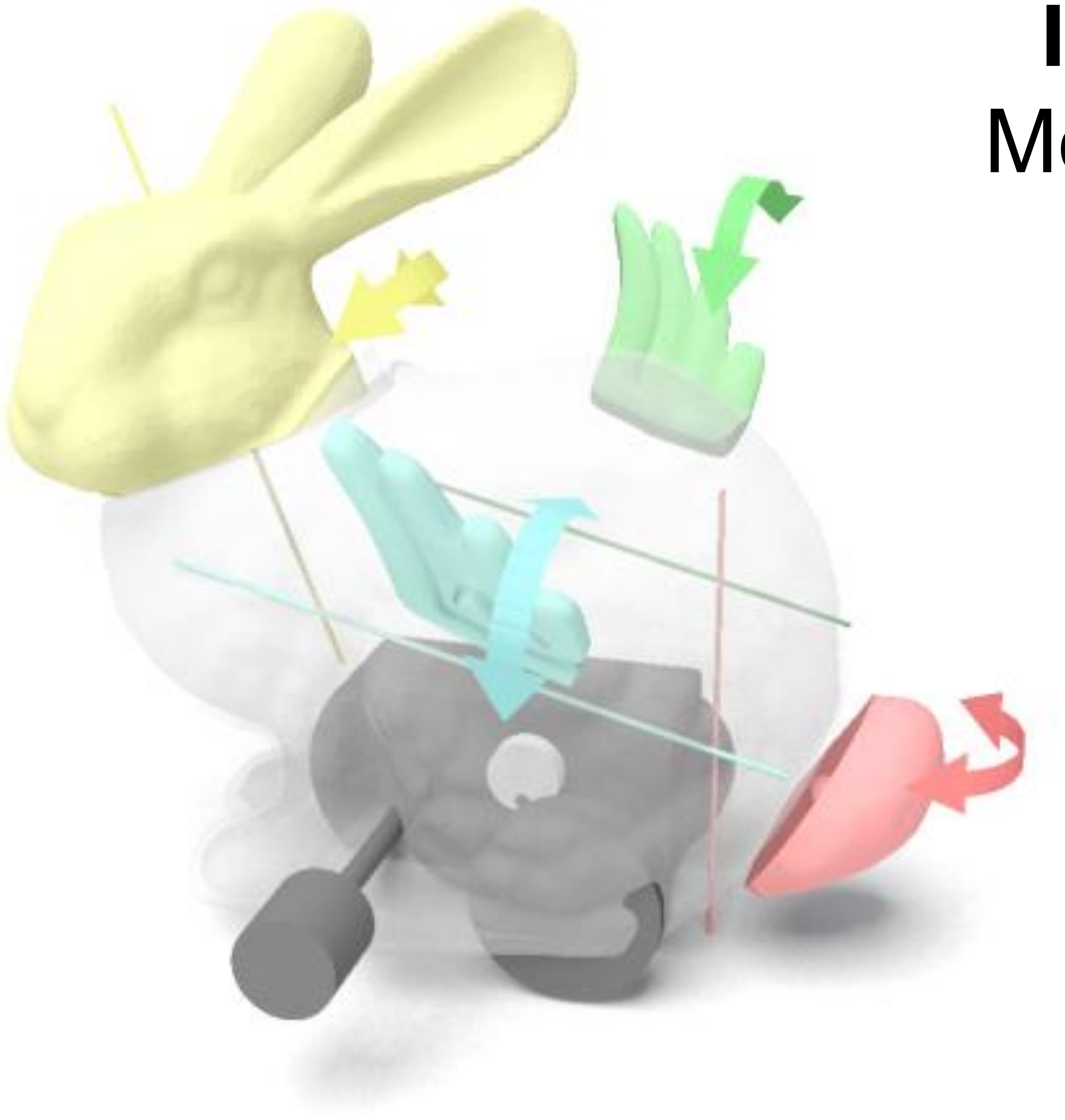
# Overview



**Input #3**

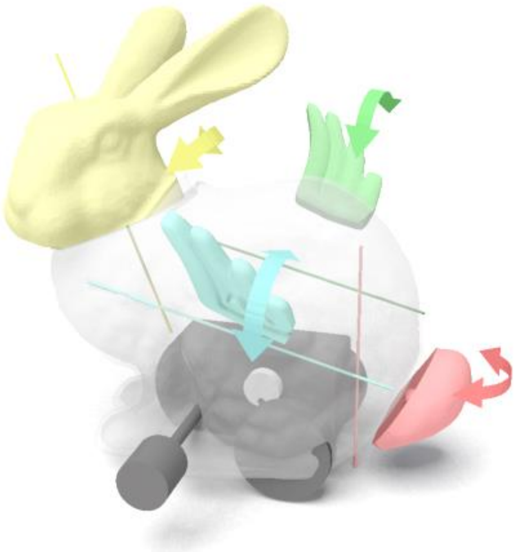
Motor Pose

# Overview



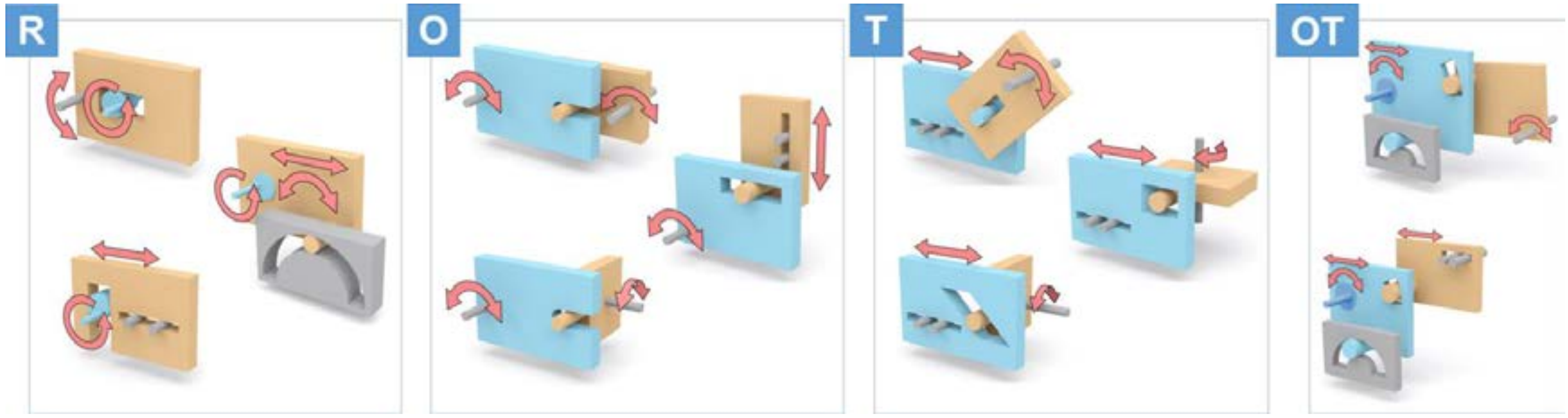
**Input #3**
Motor Pose

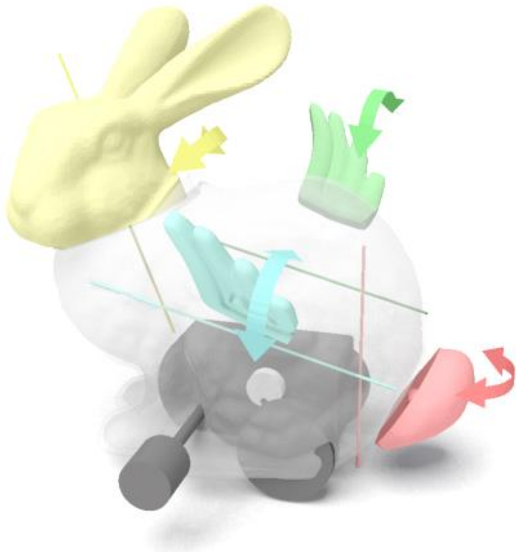
# Overview



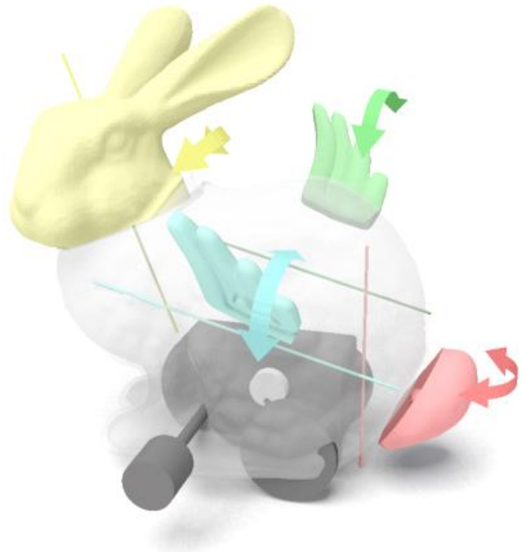
# Overview

Model elemental mechanisms

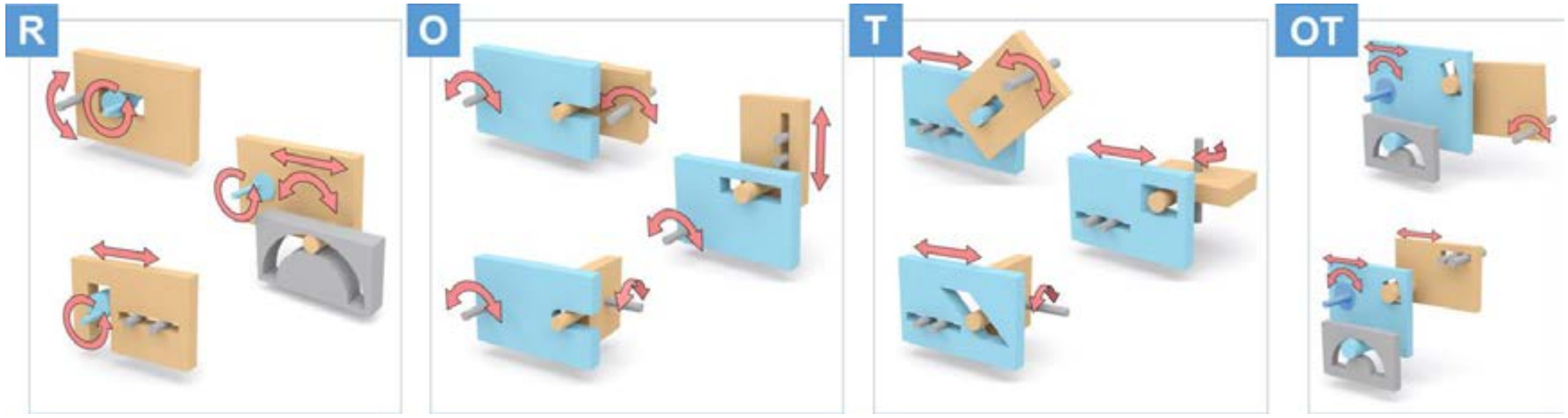


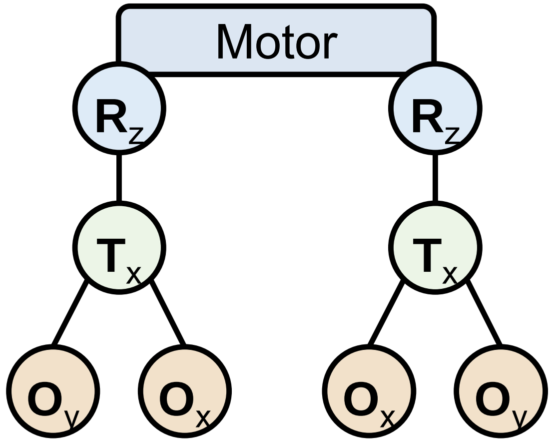
# Overview



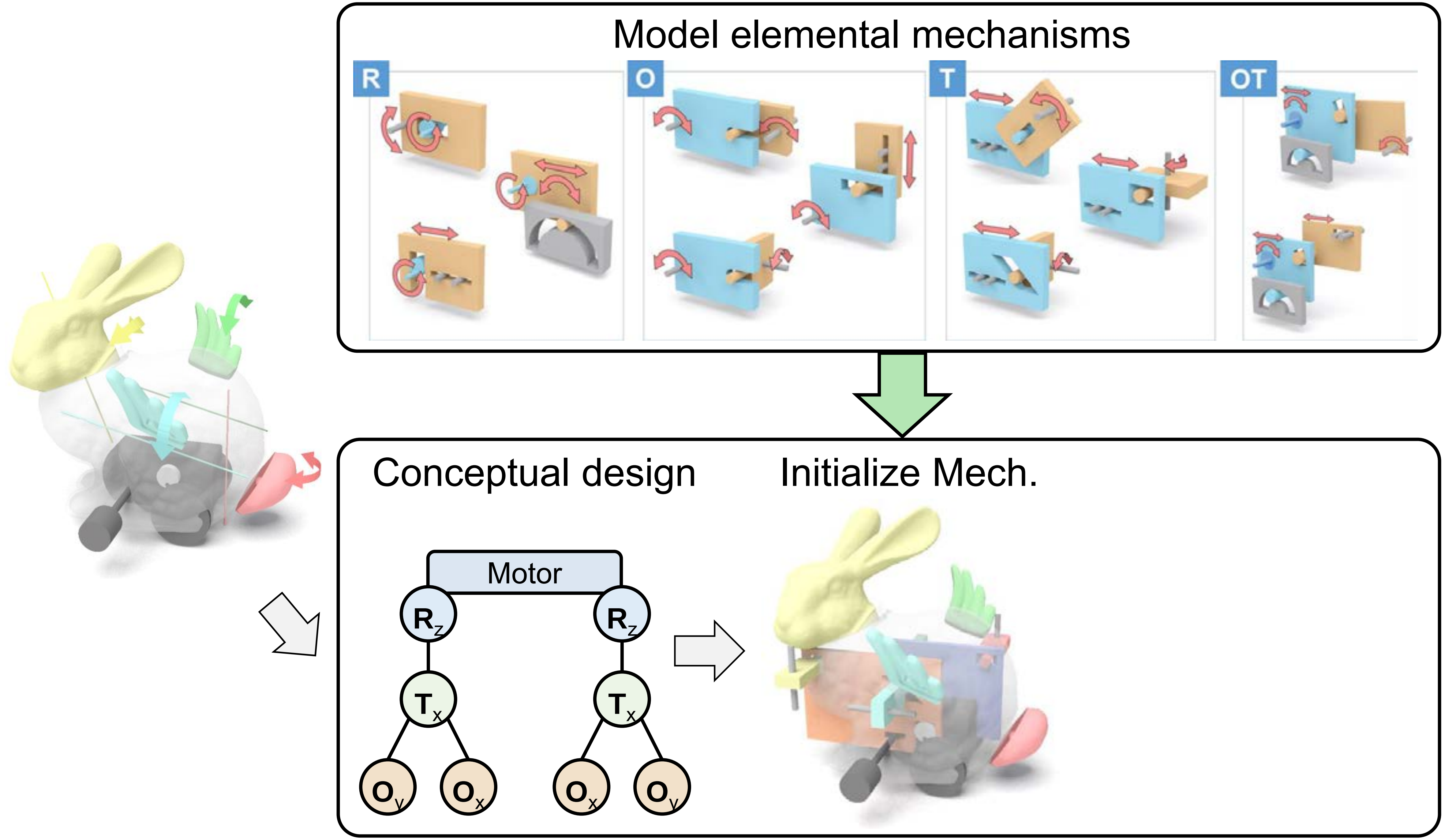
Model elemental mechanisms




Conceptual design

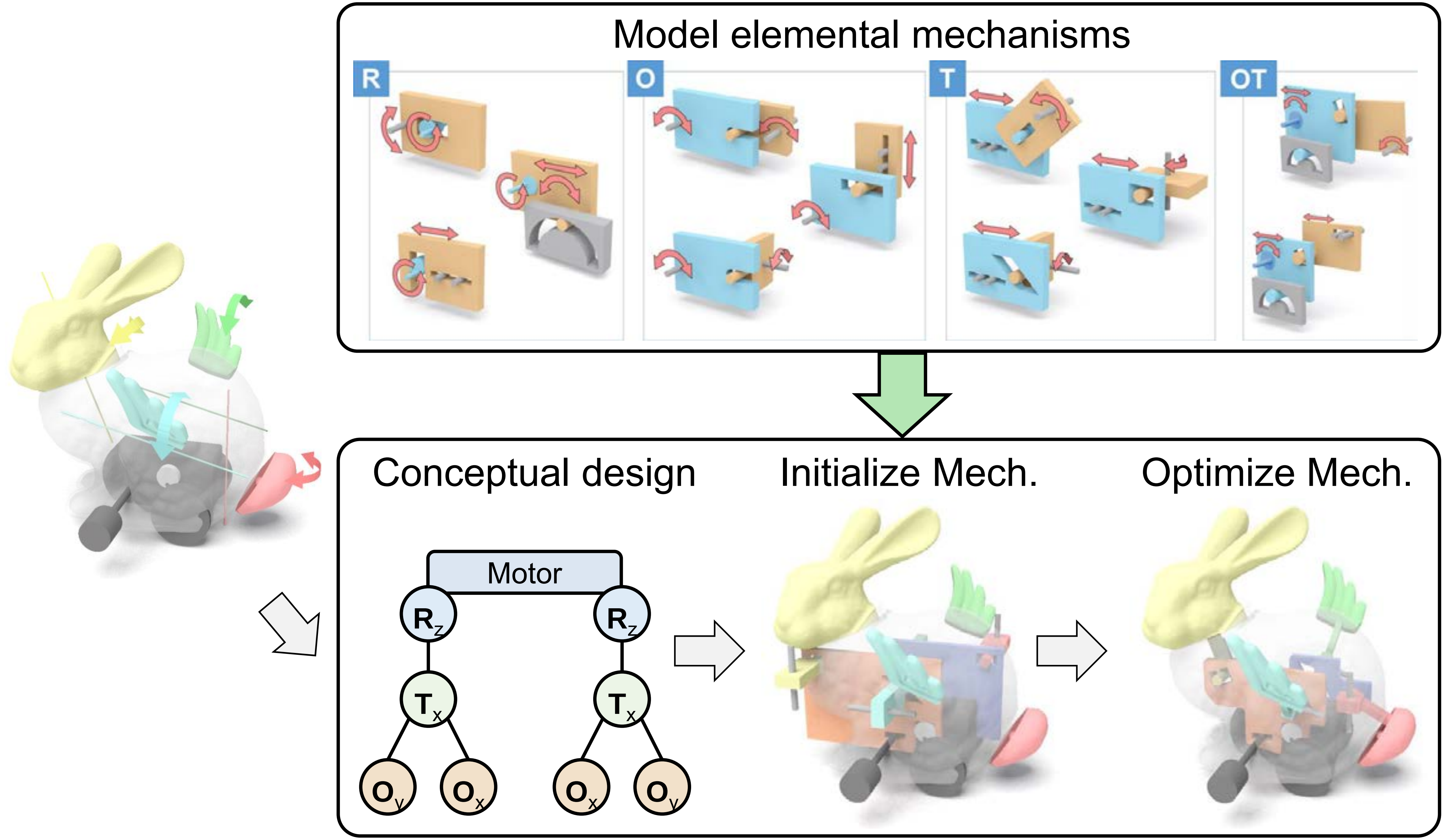


# Overview

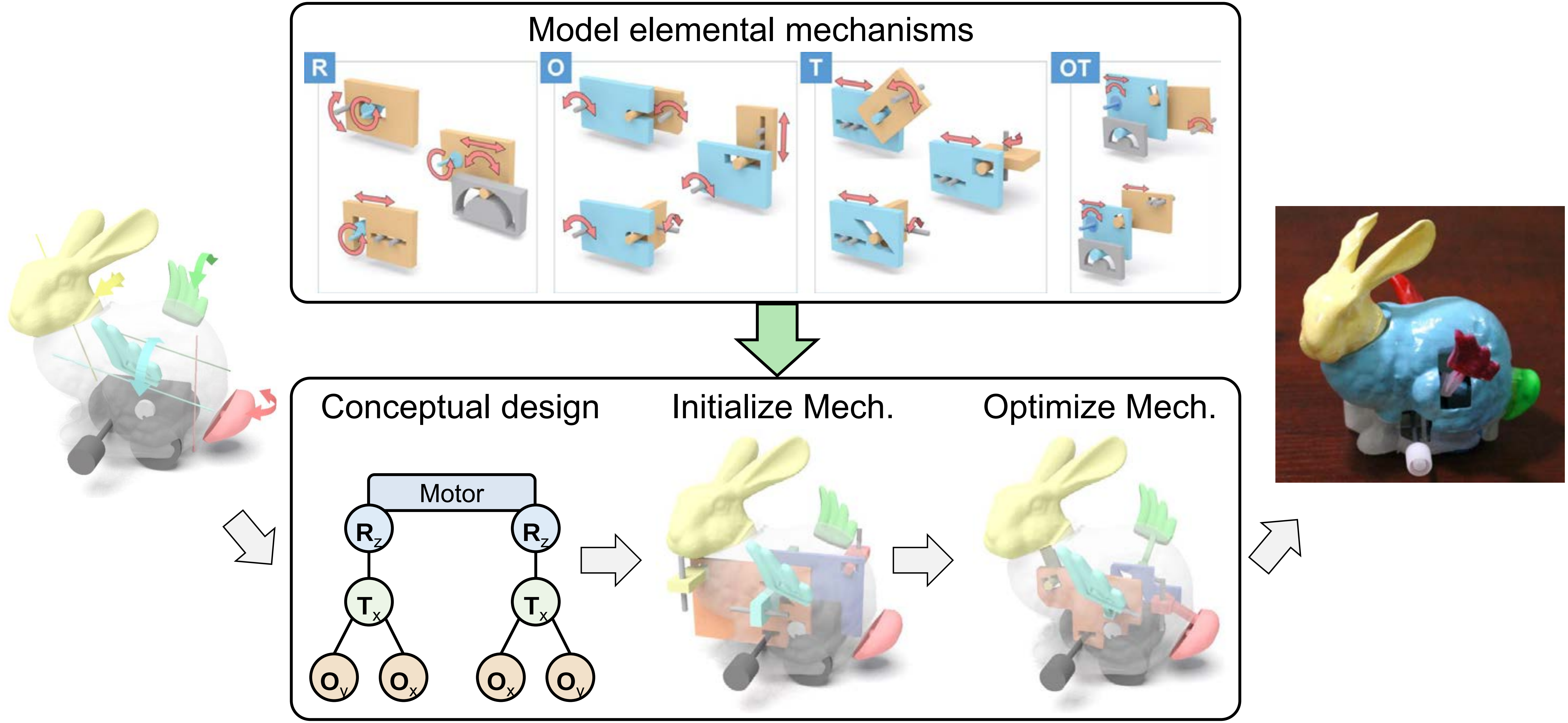


Model elemental mechanisms
R
O
T
OT
Conceptual design
Initialize Mech.
Motor
$R_z$
$R_z$
$T_x$
$T_x$
$O_y$
$O_x$
$O_x$
$O_y$

# Overview

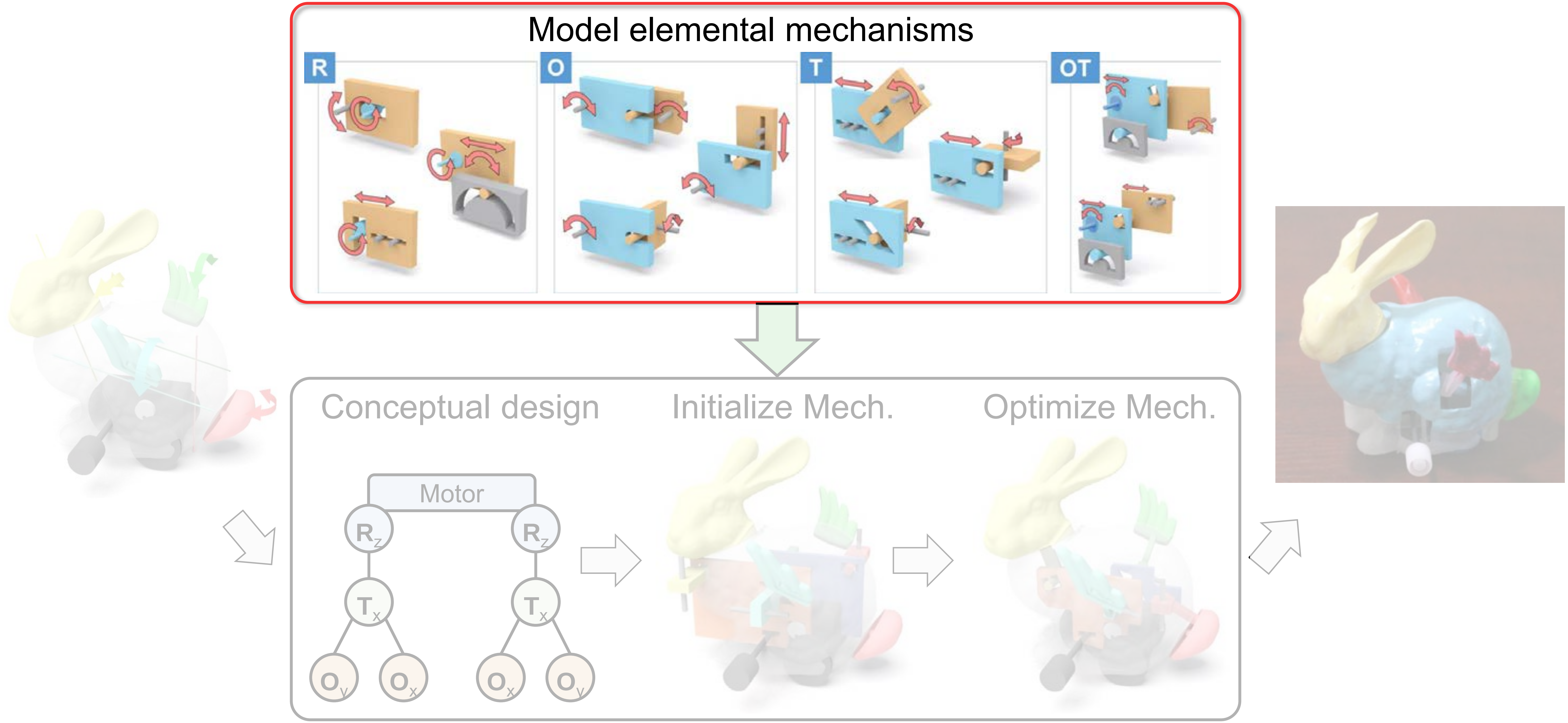

Model elemental mechanisms
R
O
T
OT
Conceptual design
Initialize Mech.
Optimize Mech.
Motor
$R_z$
$R_z$
$T_x$
$T_x$
$O_y$
$O_x$
$O_x$
$O_y$

# Overview



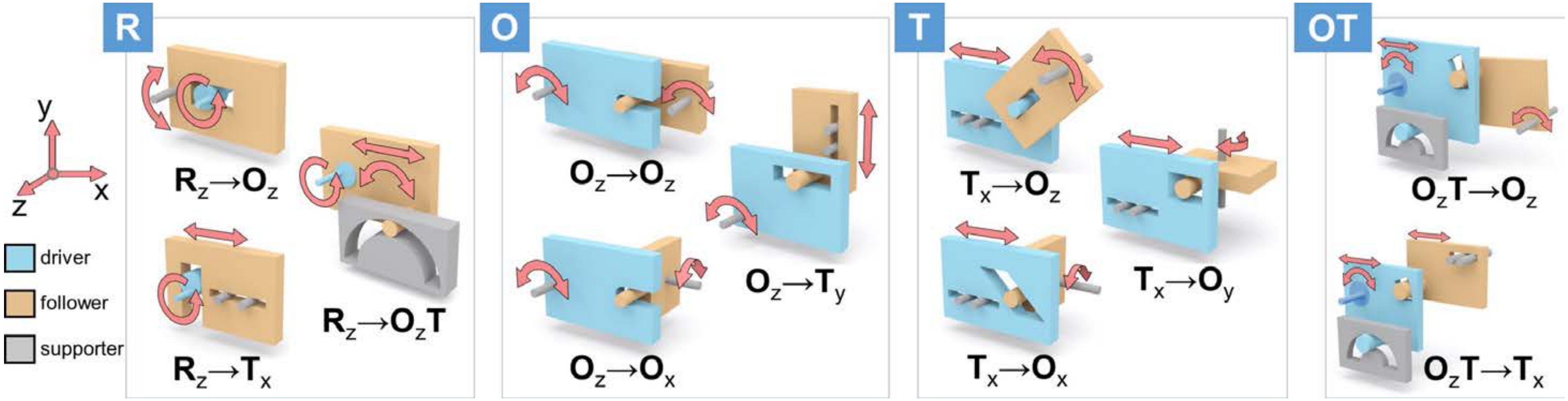
Model elemental mechanisms
R
O
T
OT
Conceptual design
Initialize Mech.
Optimize Mech.
Motor
$R_z$
$R_z$
$T_x$
$T_x$
$O_y$
$O_x$
$O_x$
$O_y$

# Overview


Model elemental mechanisms
R
O
T
OT
Conceptual design
Initialize Mech.
Optimize Mech.
Motor
$R_z$
$R_z$
$T_x$
$T_x$
$O_y$
$O_x$
$O_x$
$O_y$

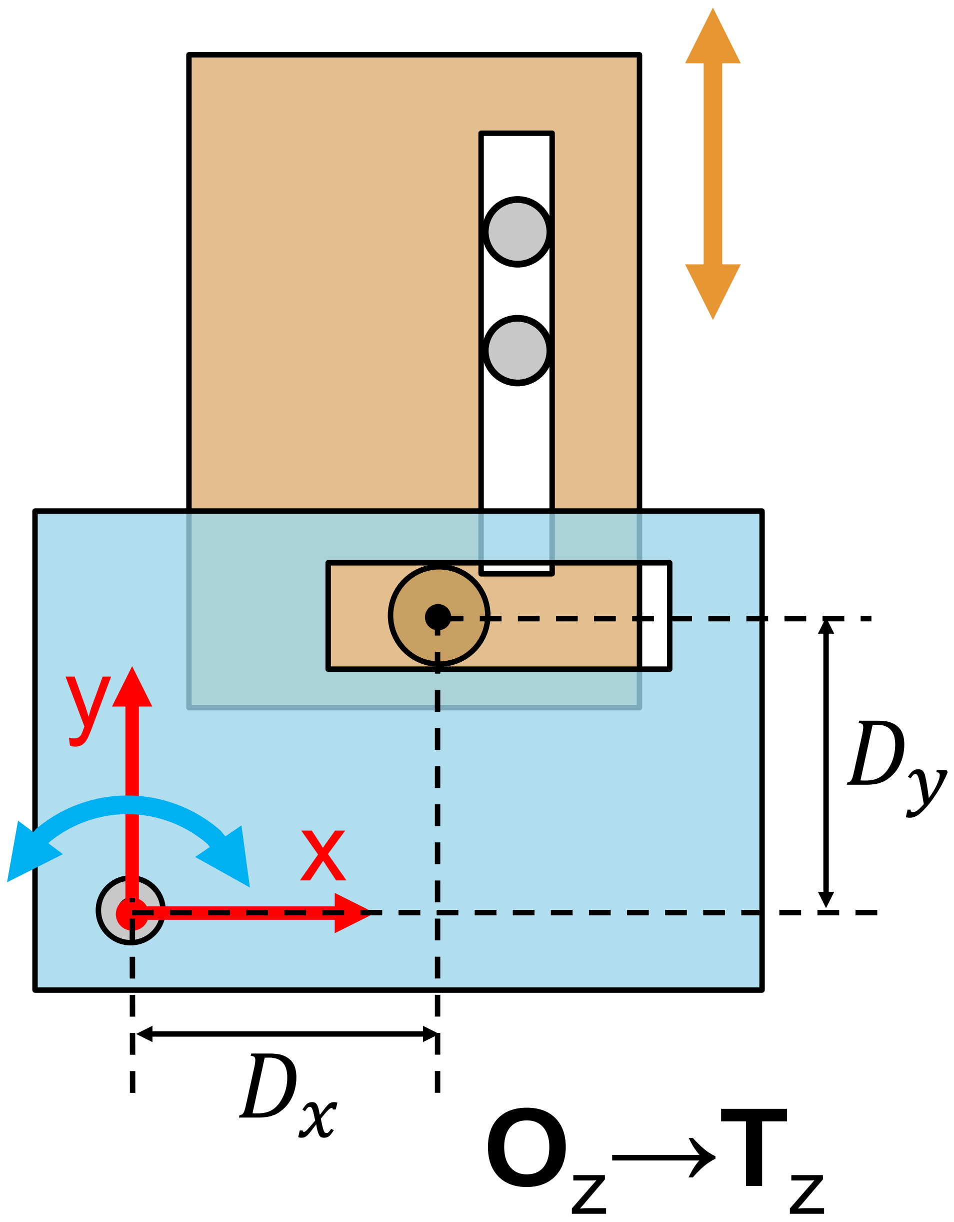
# Model Elemental Mechanisms (eleMechs)

- Each eleMech has a driver, a follower, and supporter(s)
- Categorize eleMechs into four groups: **R**, **O**, **T**, and **OT**

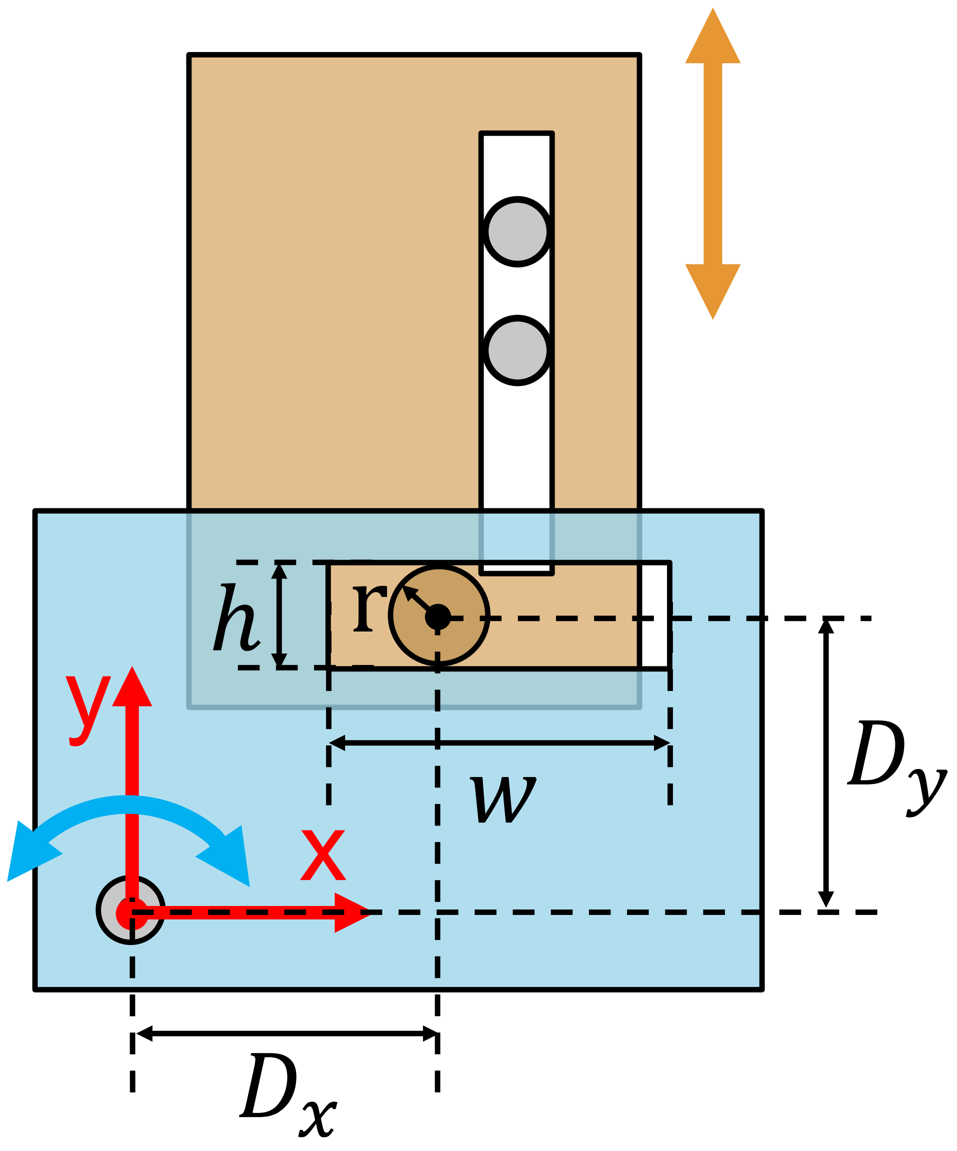


# Model Elemental Mechanisms: **Geometry**

- Parameters of joints layout

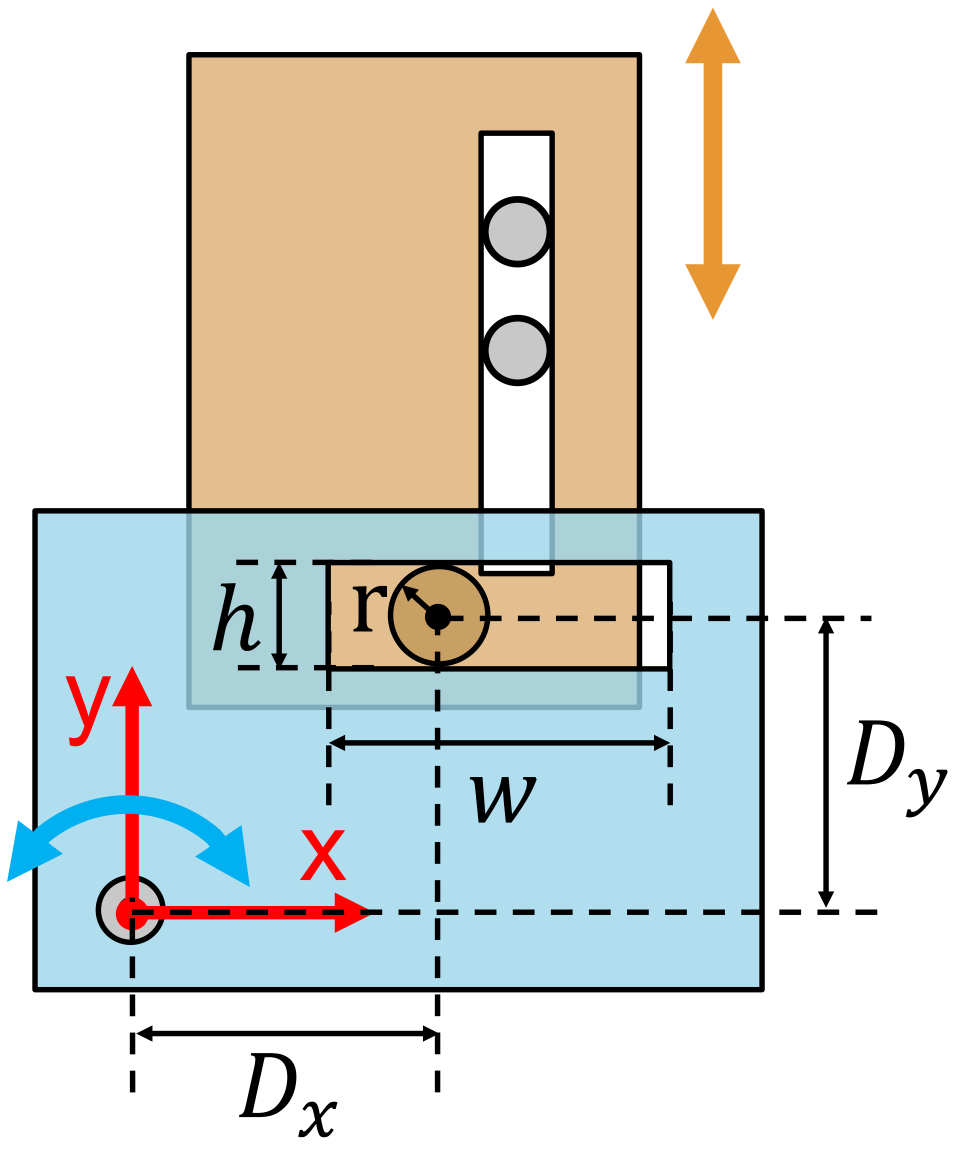



# Model Elemental Mechanisms: **Geometry**

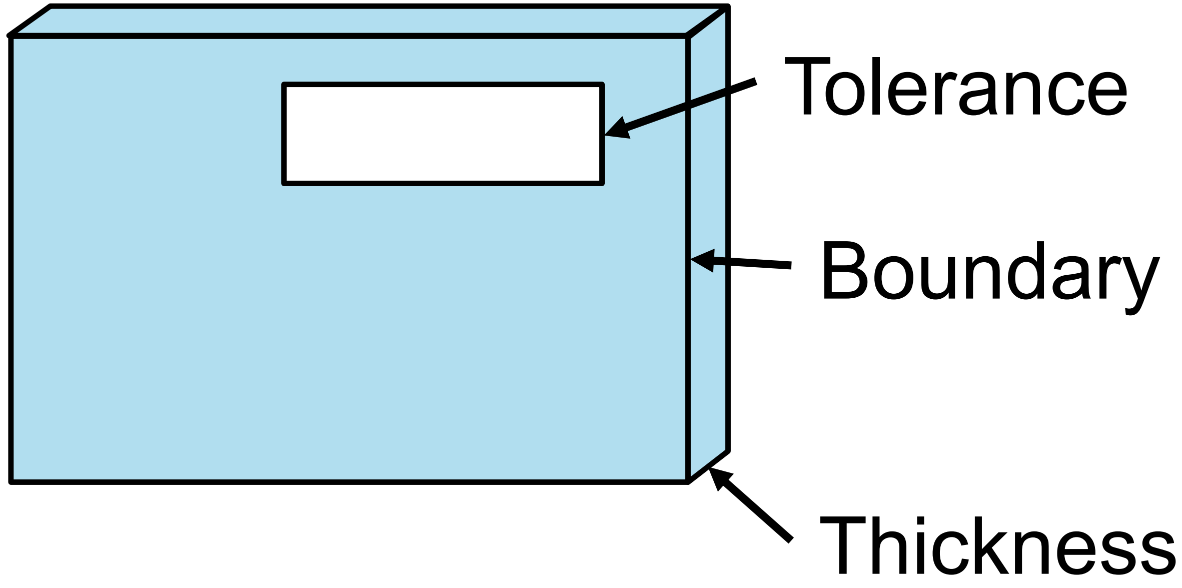



- Parameters of joints layout
- Parameters of joints geometry

# Model Elemental Mechanisms: **Geometry**



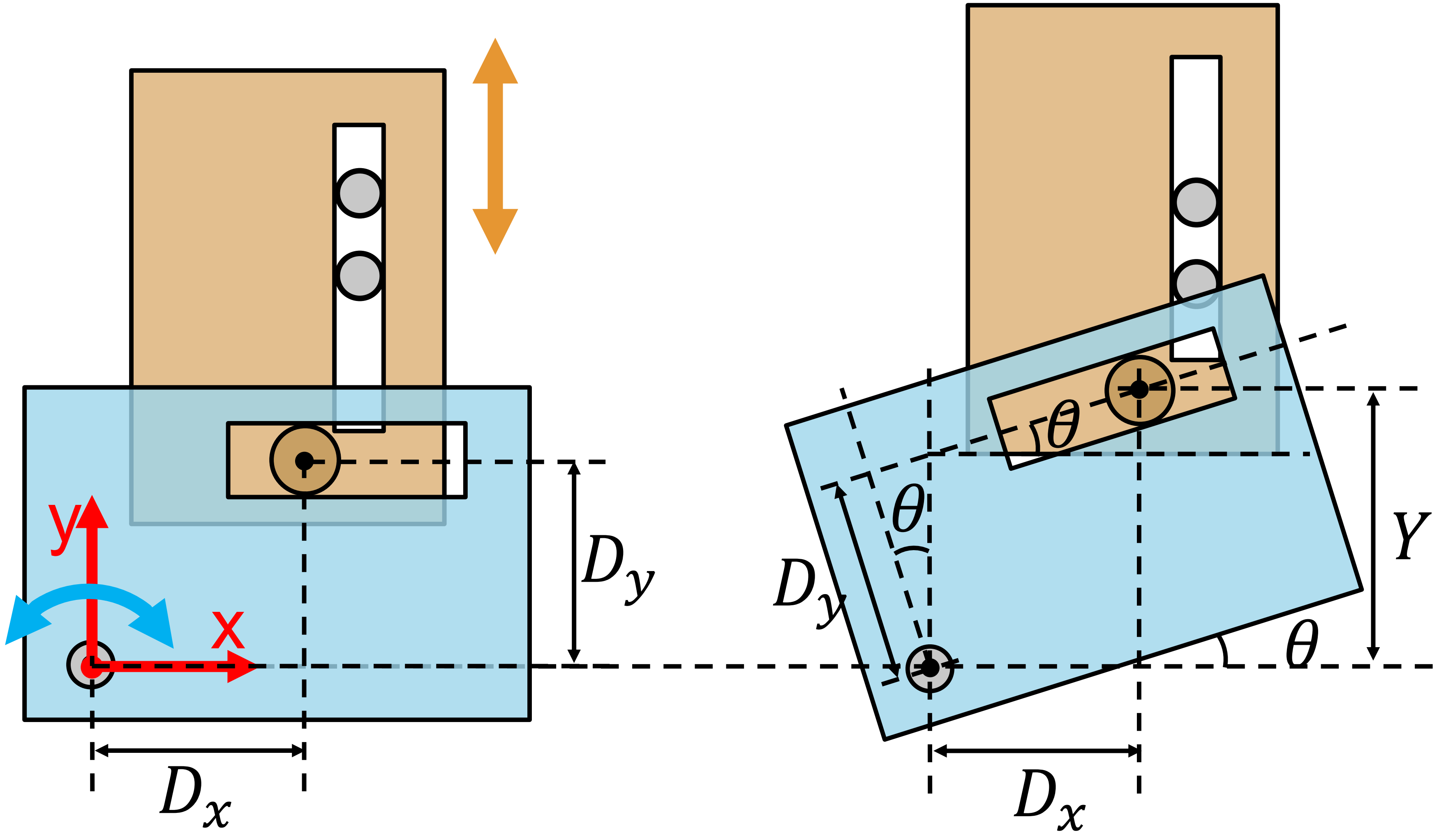

- Parameters of joints layout
- Parameters of joints
- Parameters for fabrication



# Model Elemental Mechanisms: **Kinematics**

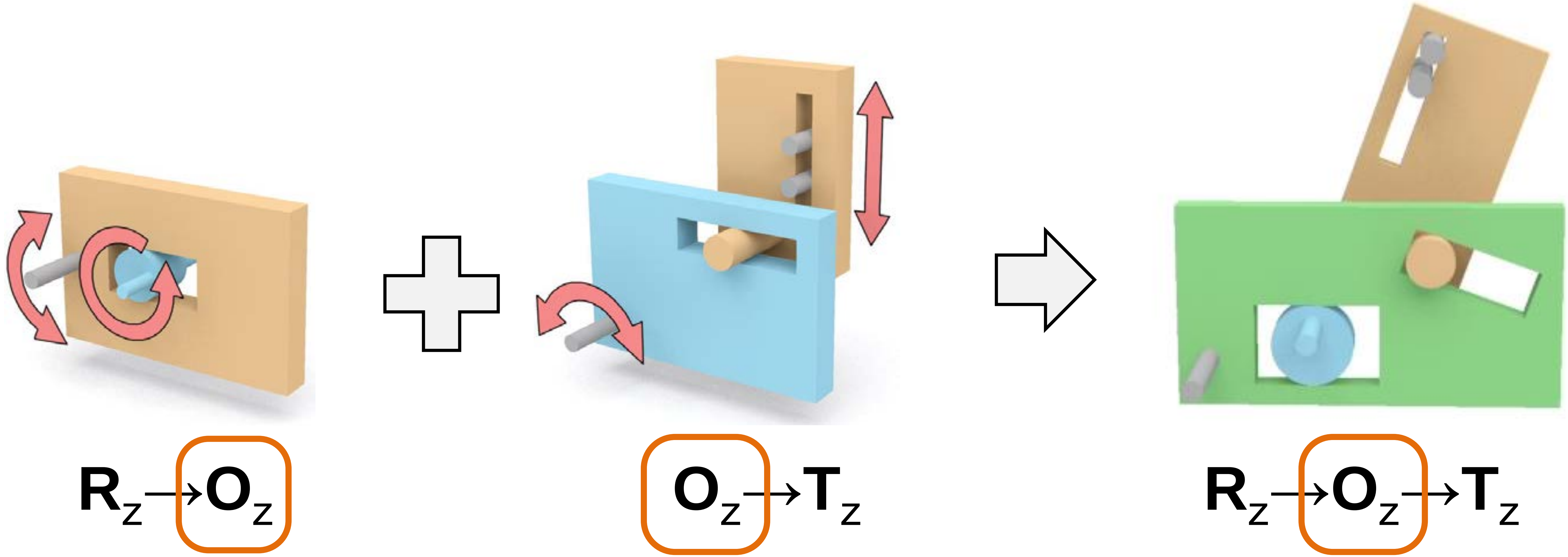
Given driver's pose at time t, compute follower's pose from the joints parameters

$$Y = \frac{D_y}{\cos\theta} + D_x \tan\theta$$

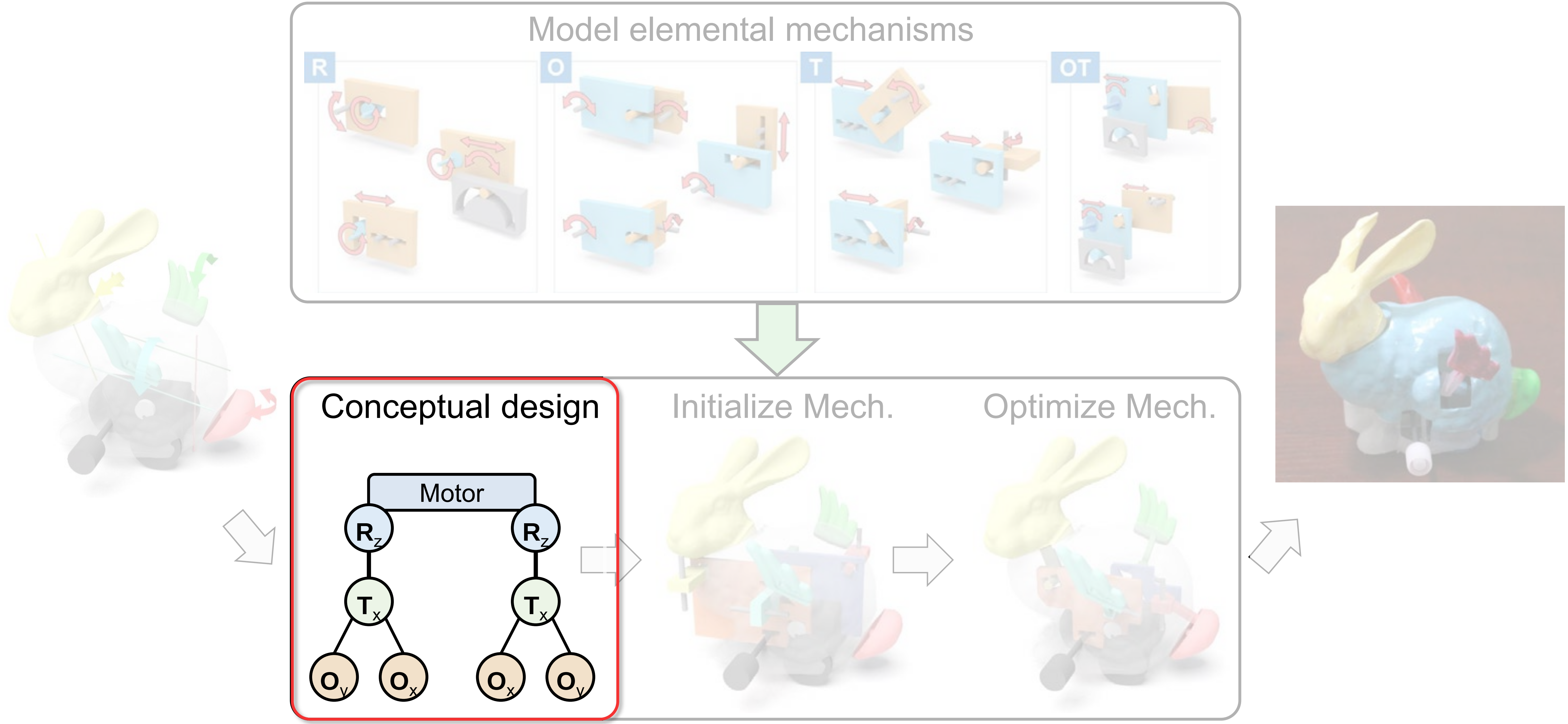


# Model Elemental Mechanisms: **Connection**

**Condition**: motion type of parent eleMech's follower should be the same as that of child eleMech's driver



# Overview



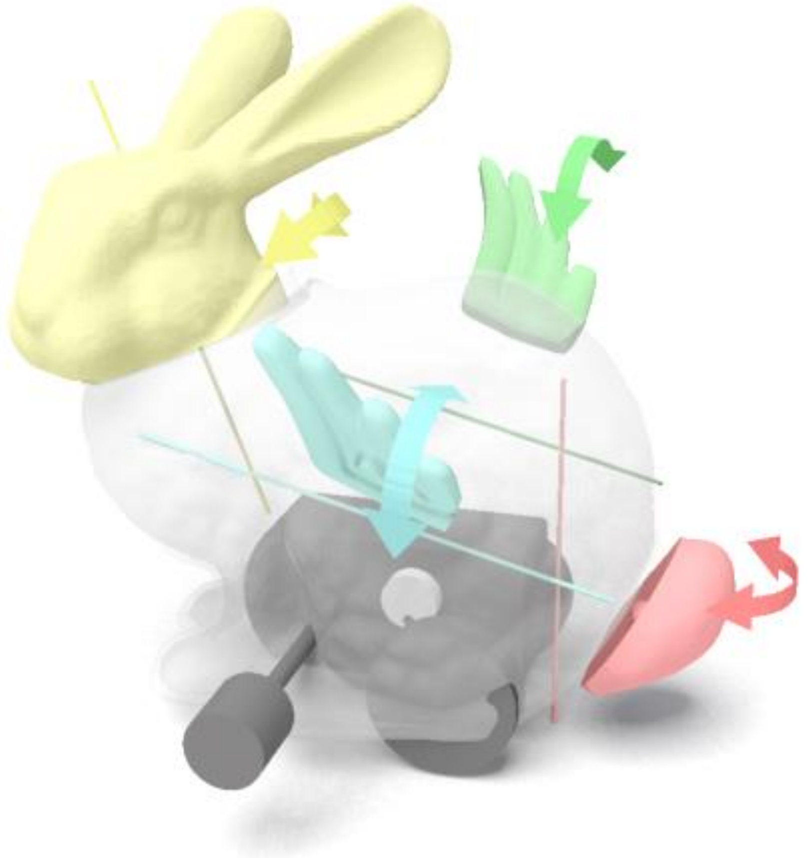
Model elemental mechanisms
R
O
T
OT
Conceptual design
Motor
$R_z$
$R_z$
$T_x$
$T_x$
$O_y$
$O_x$
$O_x$
$O_y$
Initialize Mech.
Optimize Mech.

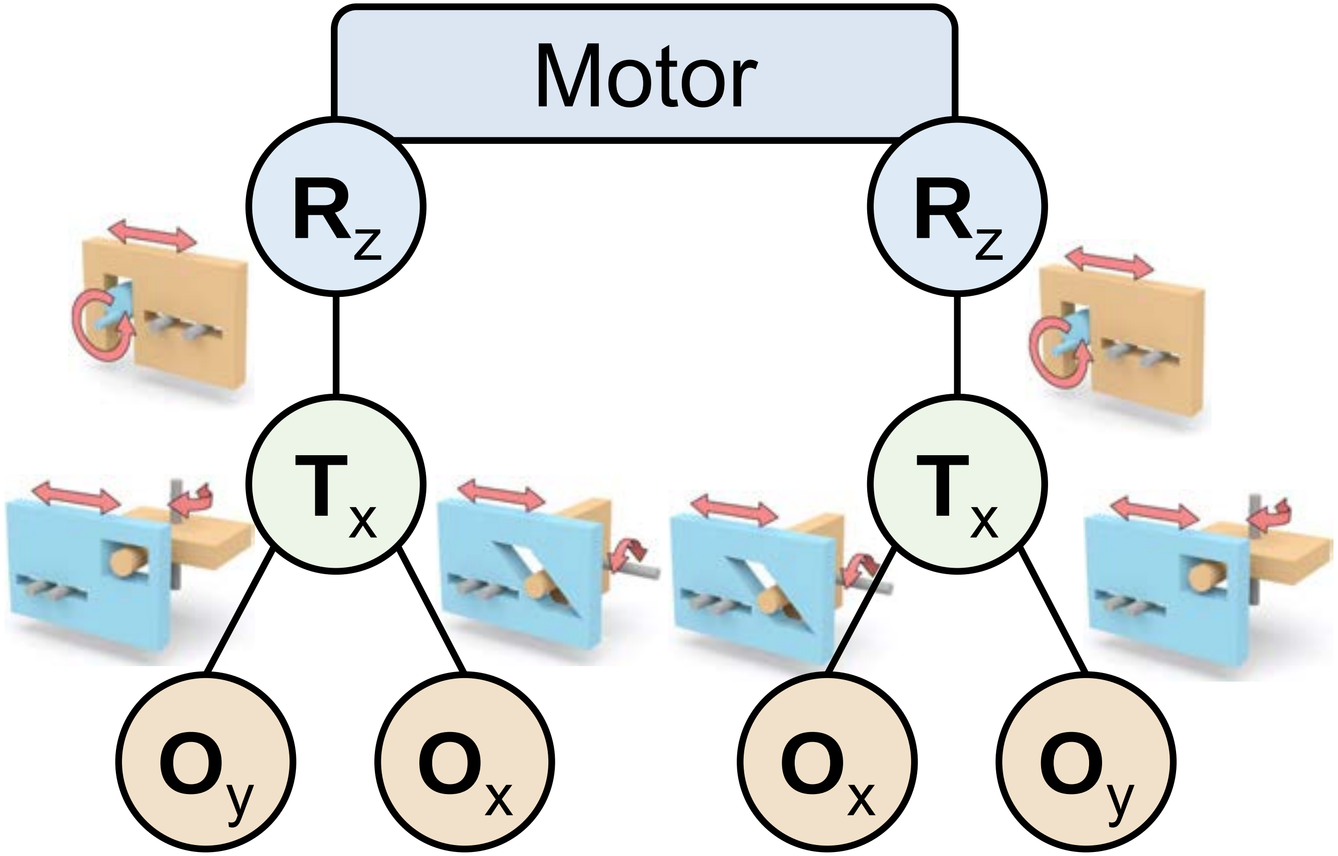
# Conceptual Design

Given user inputs and eleMech table, this stage addresses two questions:

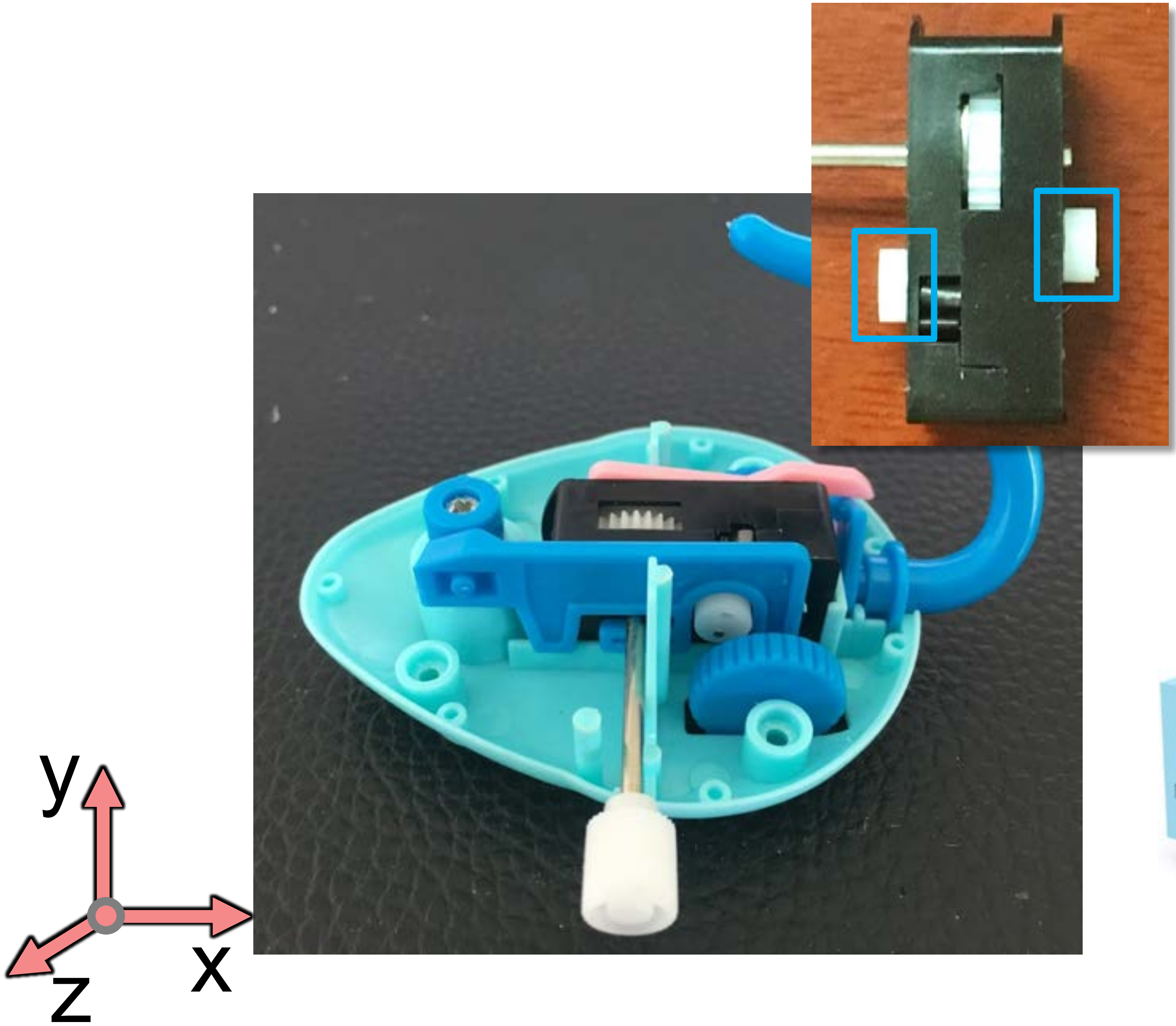
1) Which eleMechs should be selected?

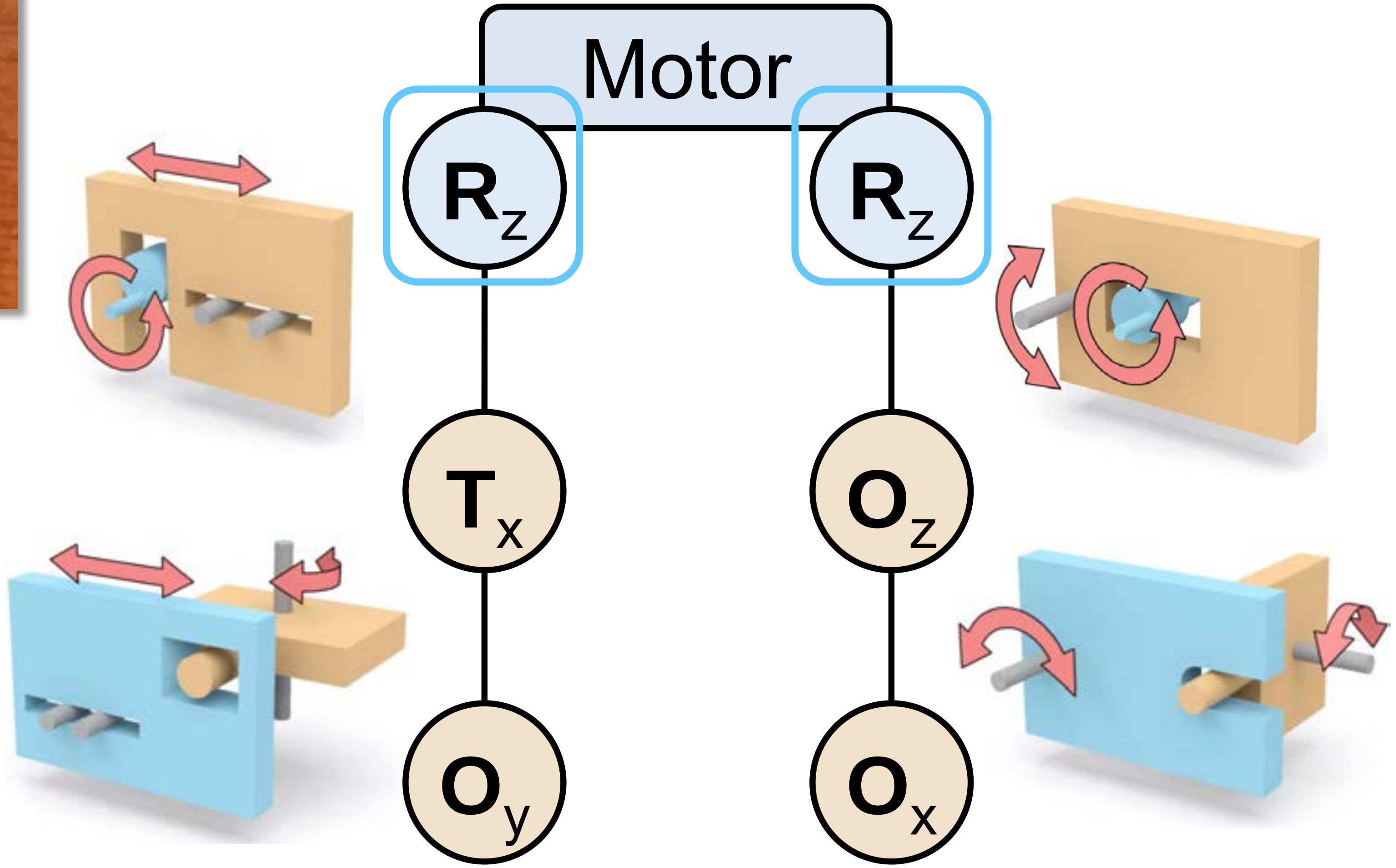
2) How these eleMechs should be connected?



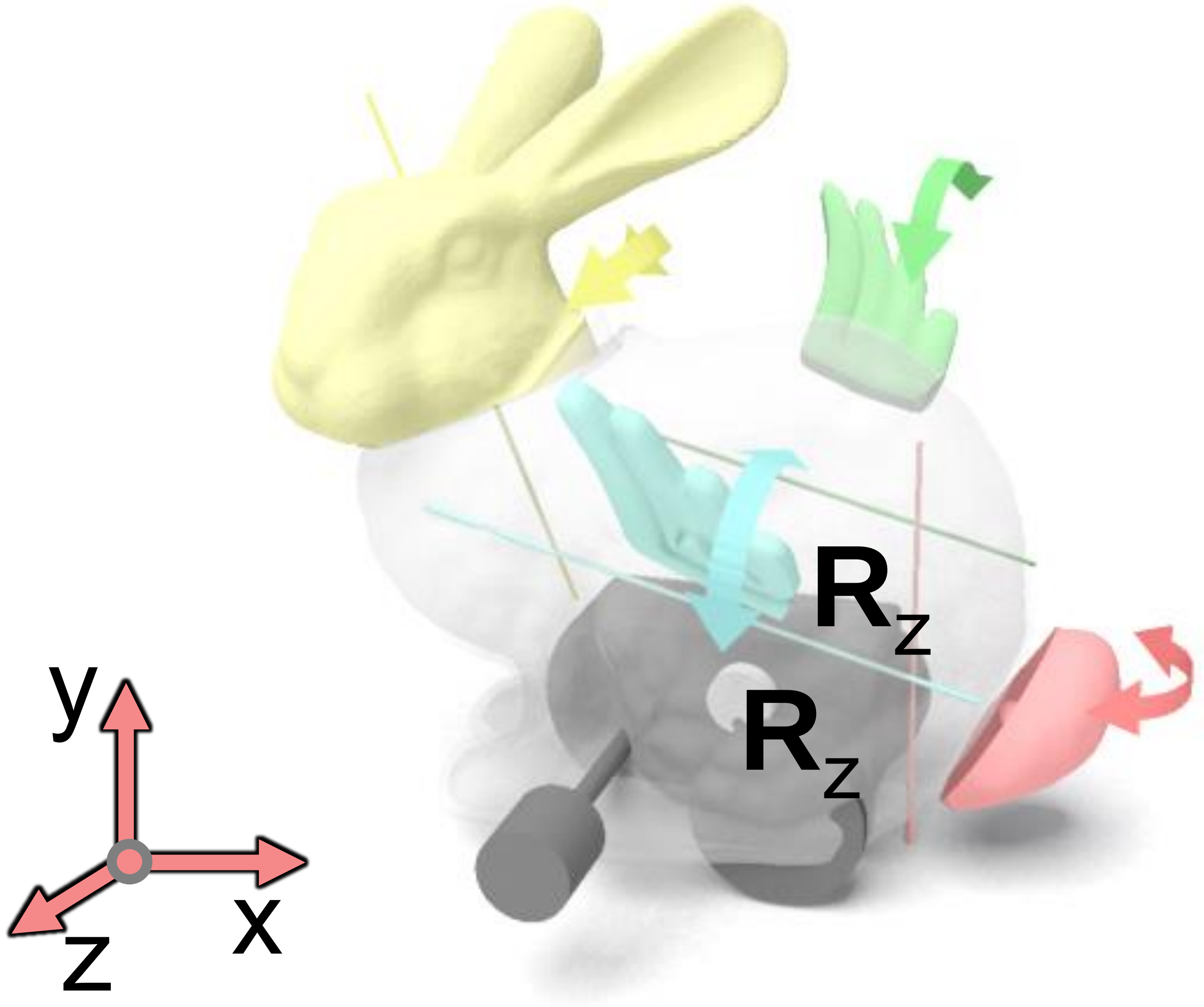


# Conceptual Design: **Motion Transfer Tree**

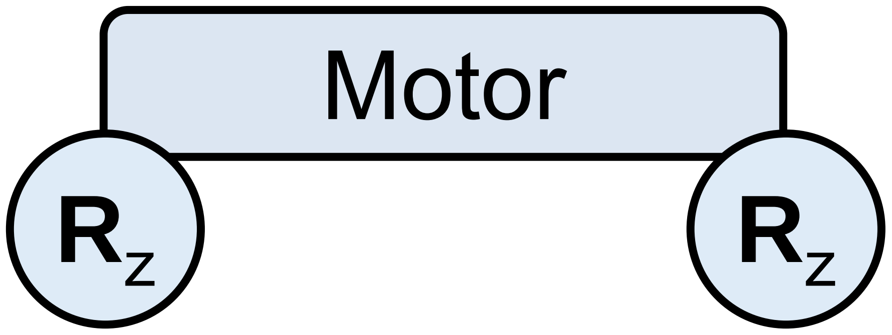


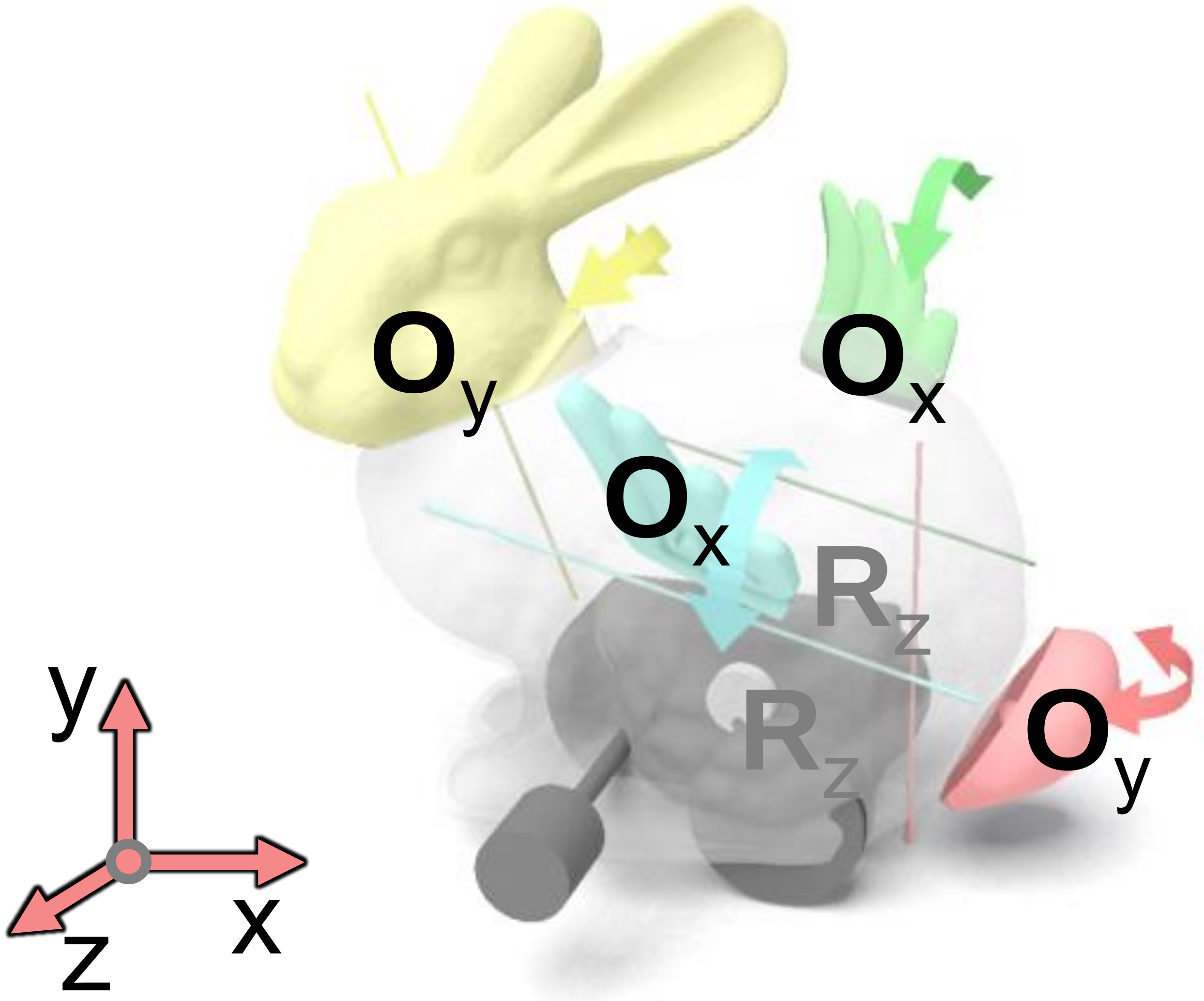
# Conceptual Design: **Construct the Tree**

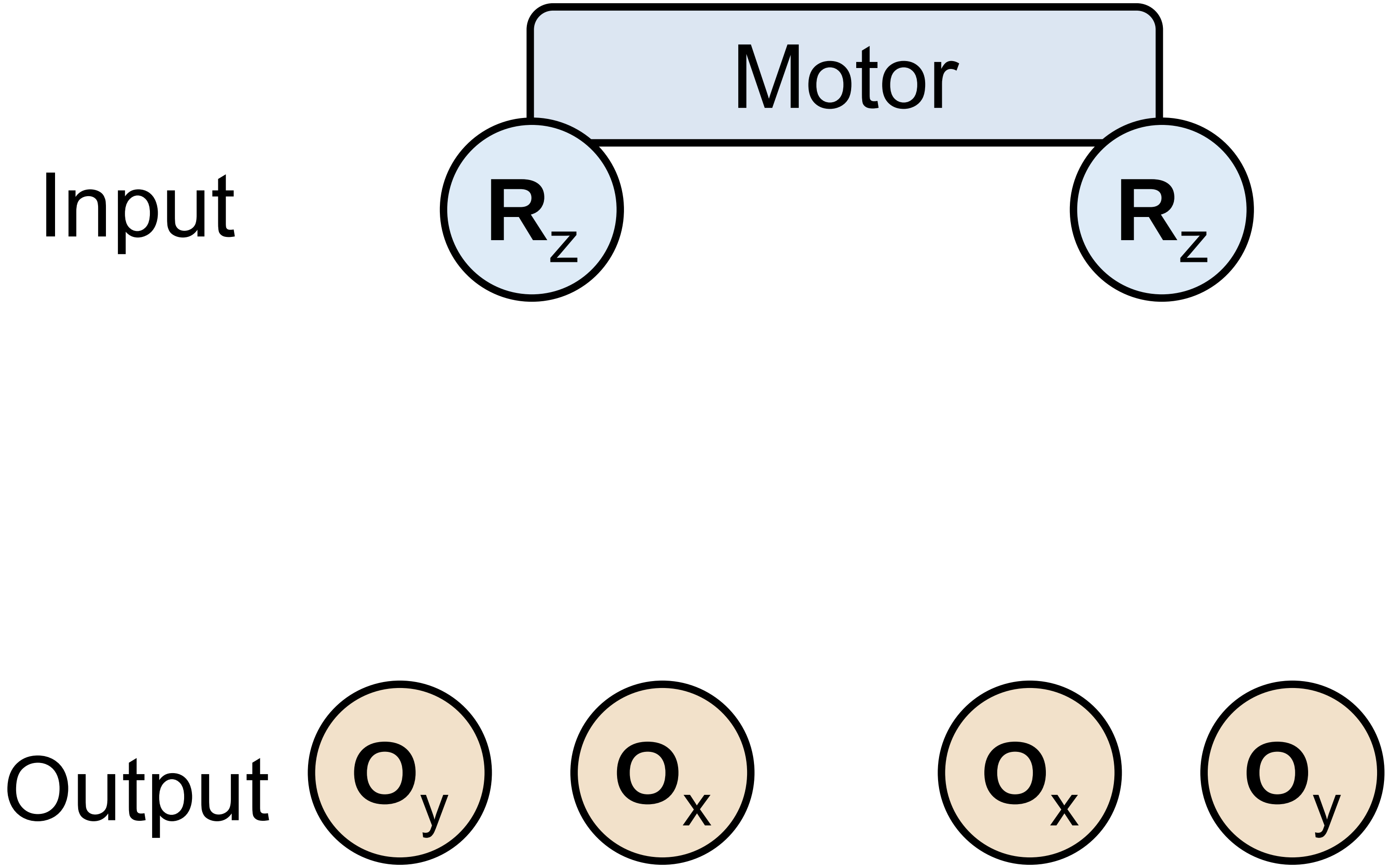



Input

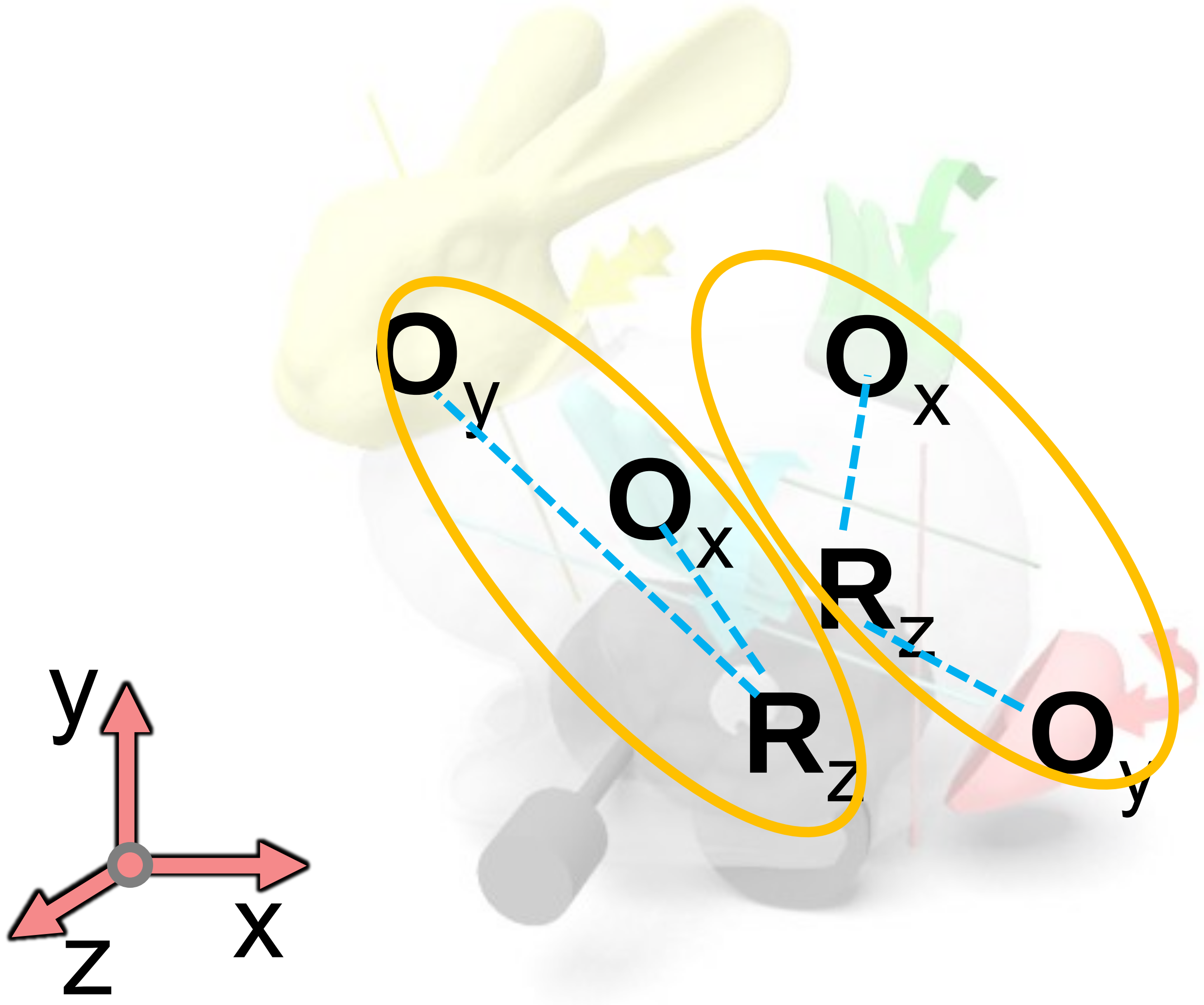


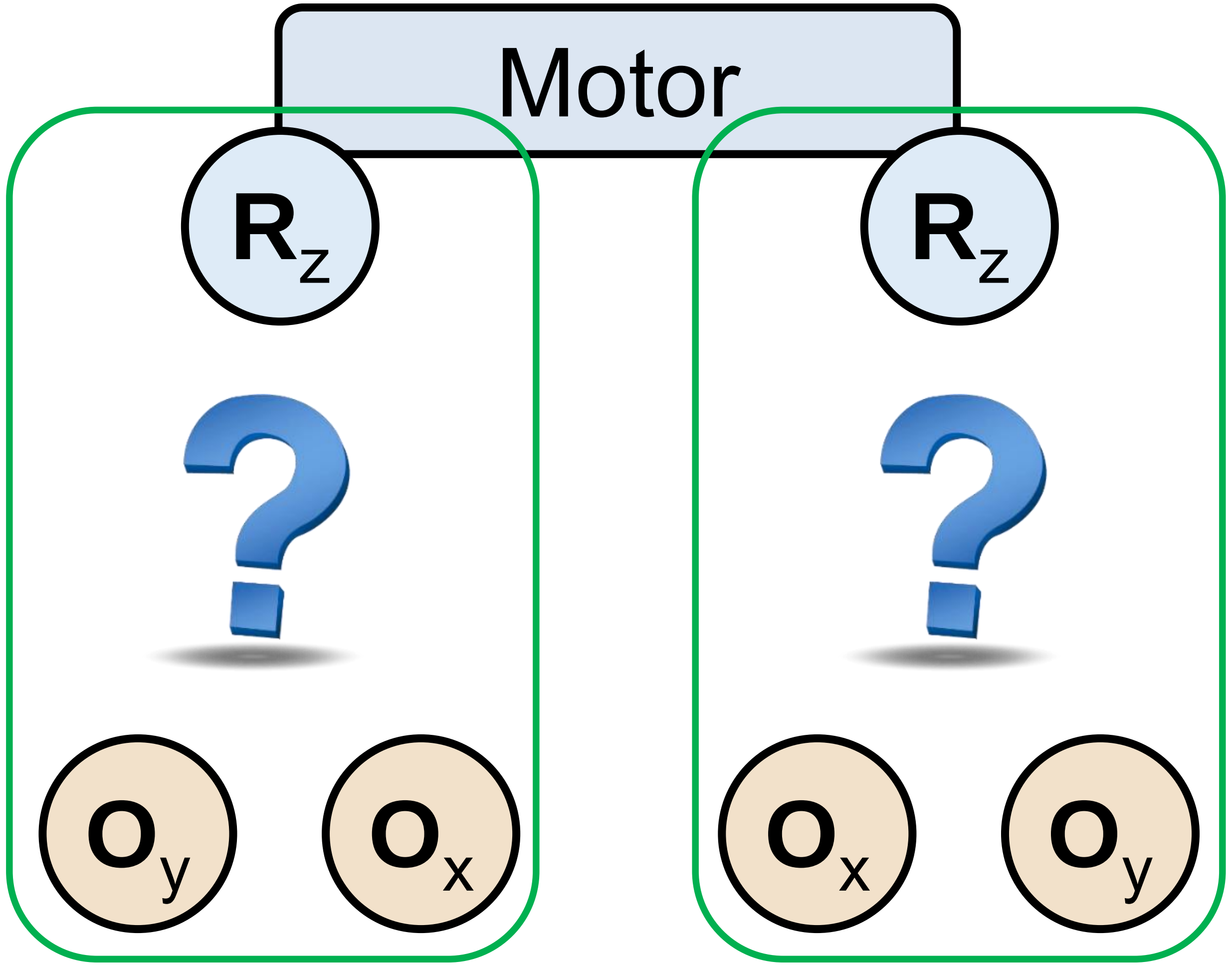
# Conceptual Design: **Construct the Tree**



# Conceptual Design: **Construct the Tree**

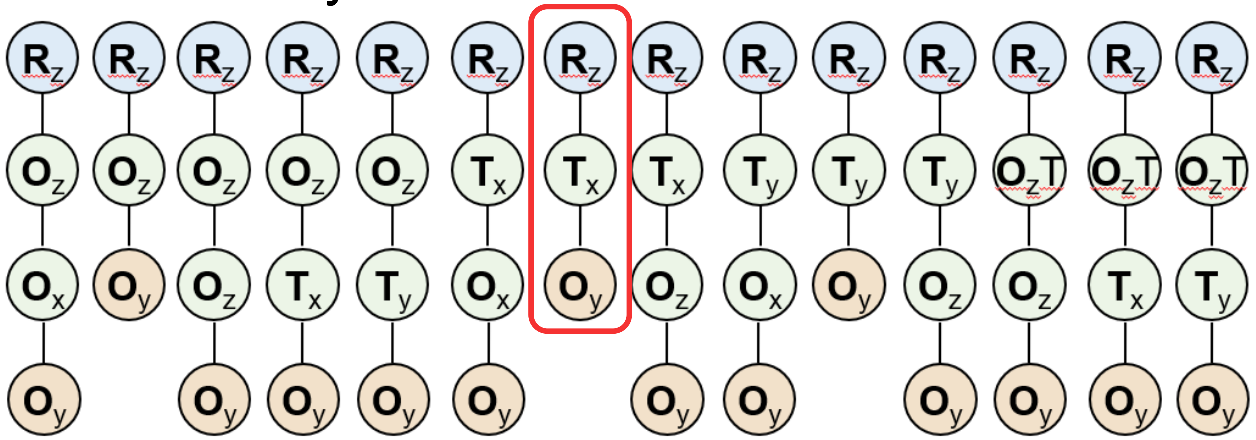


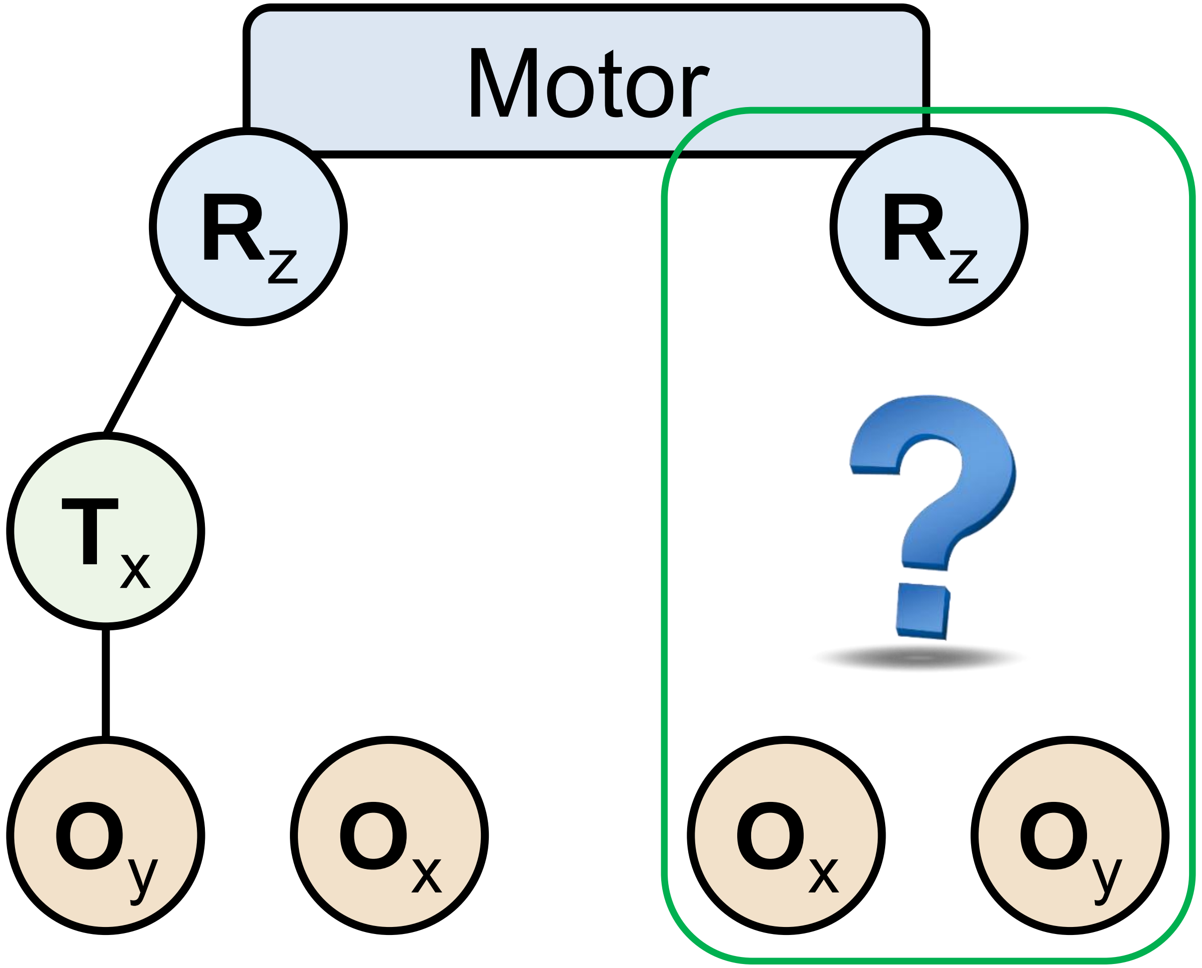




end effectors assigned to left and right cams

# Conceptual Design: **Construct the Tree**

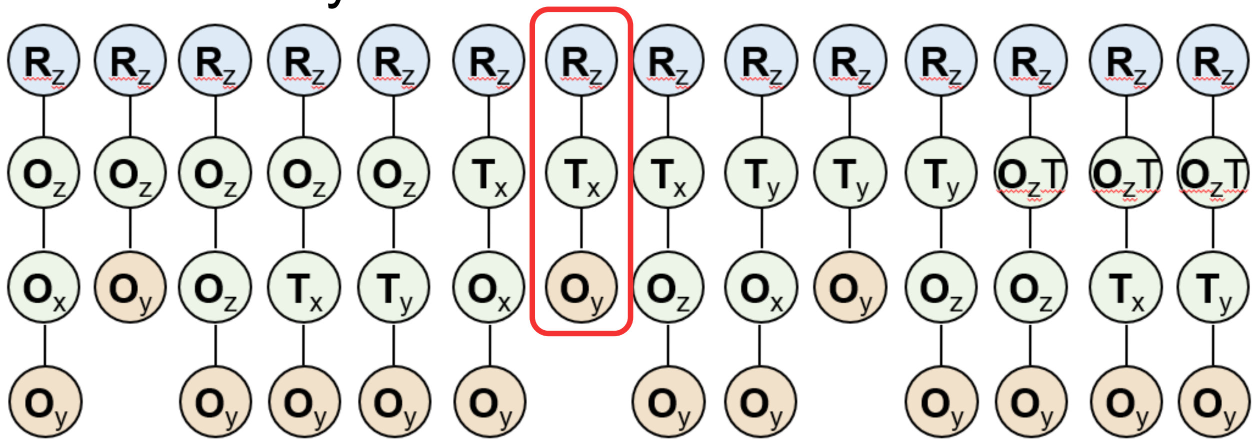
**R**$_z$ to **O**$_y$: 14 chains



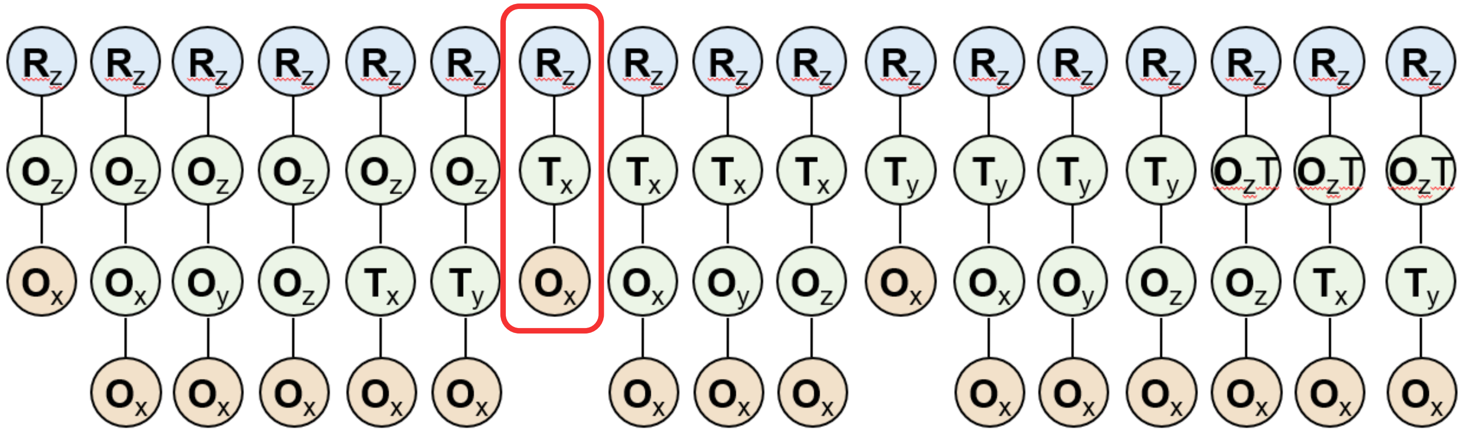


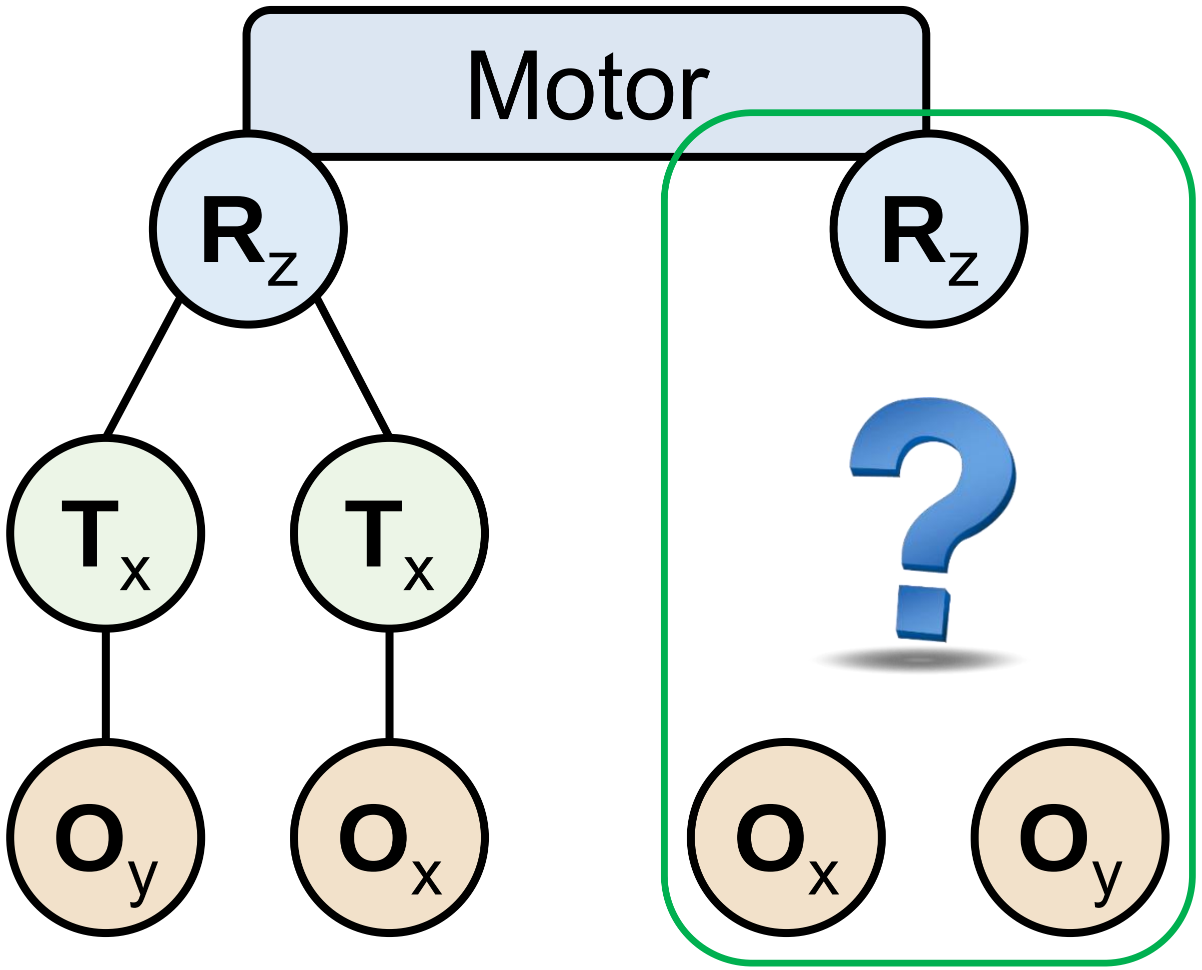
# Conceptual Design: **Construct the Tree**

**$R_z$** to **$O_y$**: 14 chains



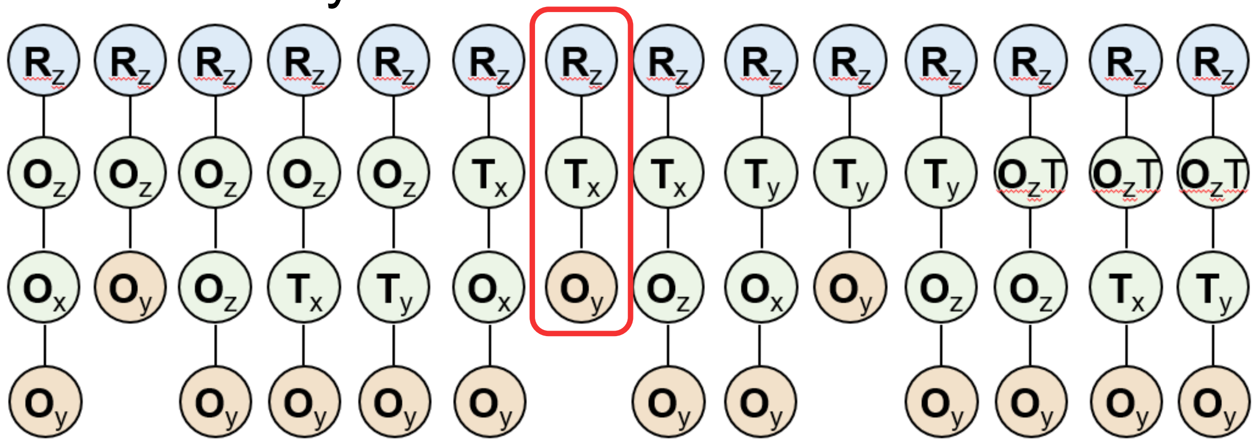

**$R_z$** to **$O_x$**: 17 chains



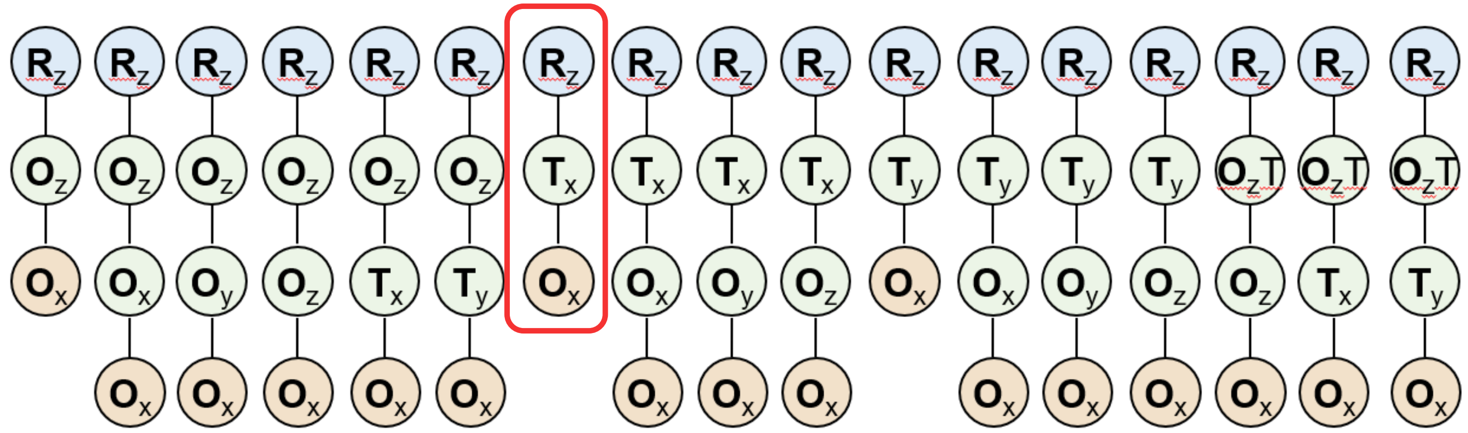



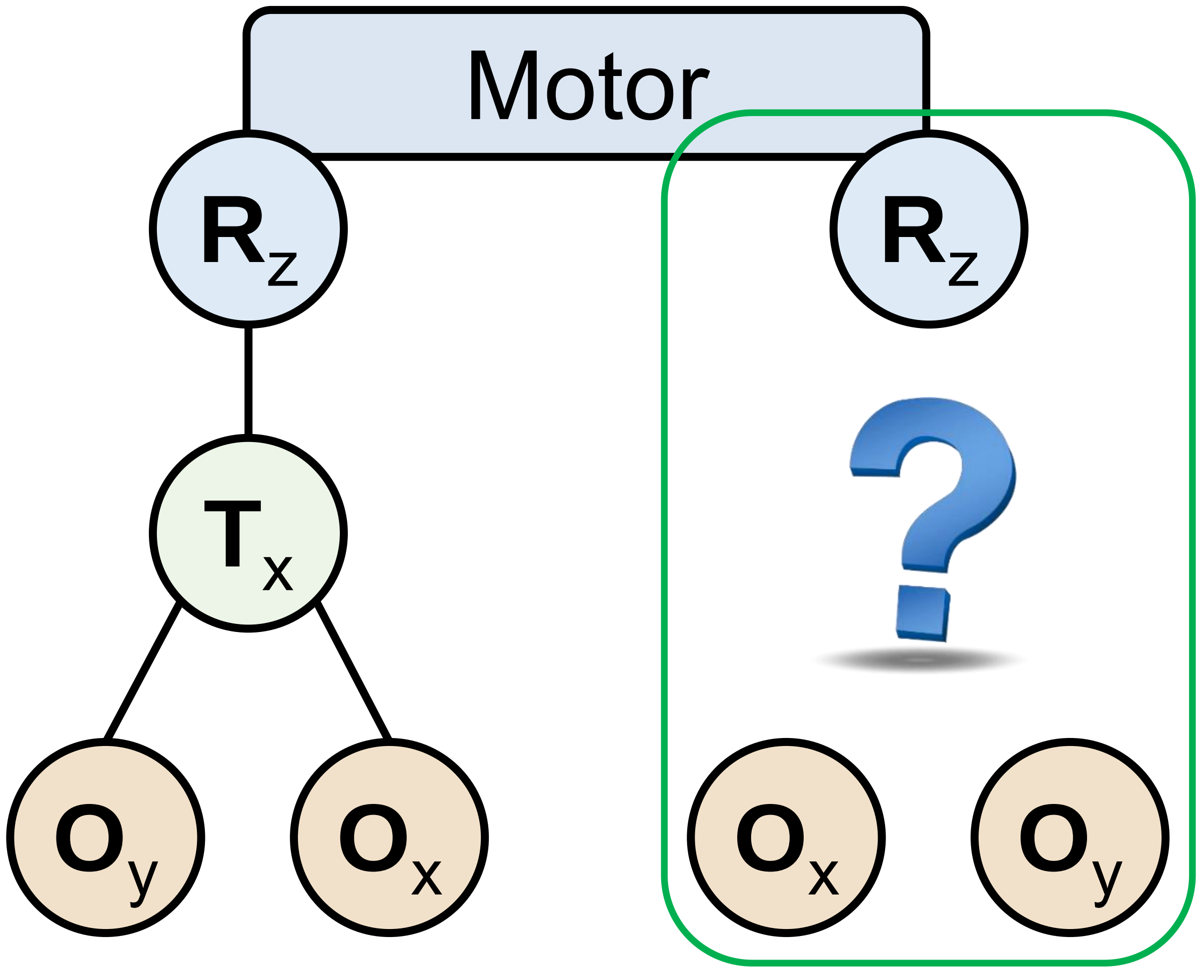
# Conceptual Design: **Construct the Tree**

**$R_z$** to **$O_y$**: 14 chains

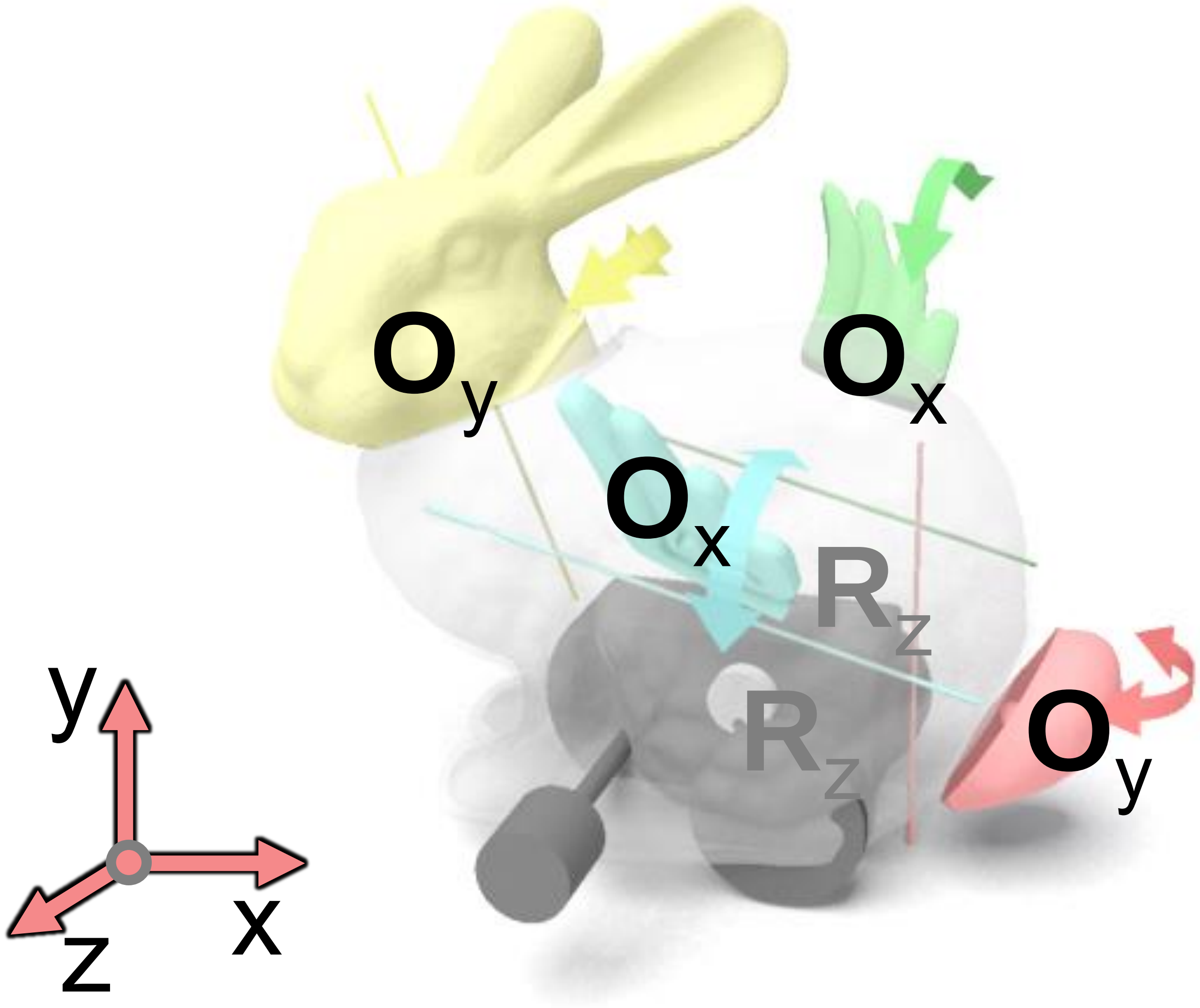


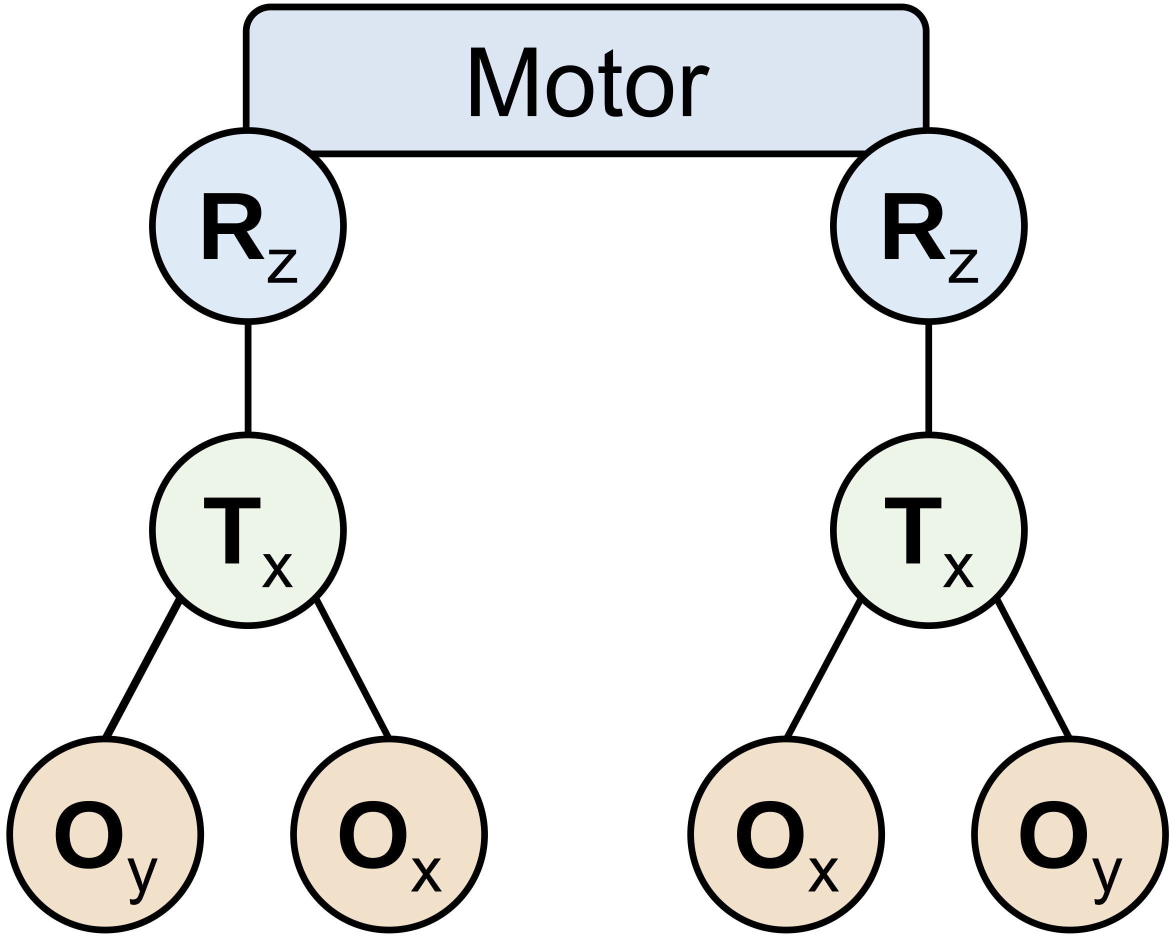

**$R_z$** to **$O_x$**: 17 chains



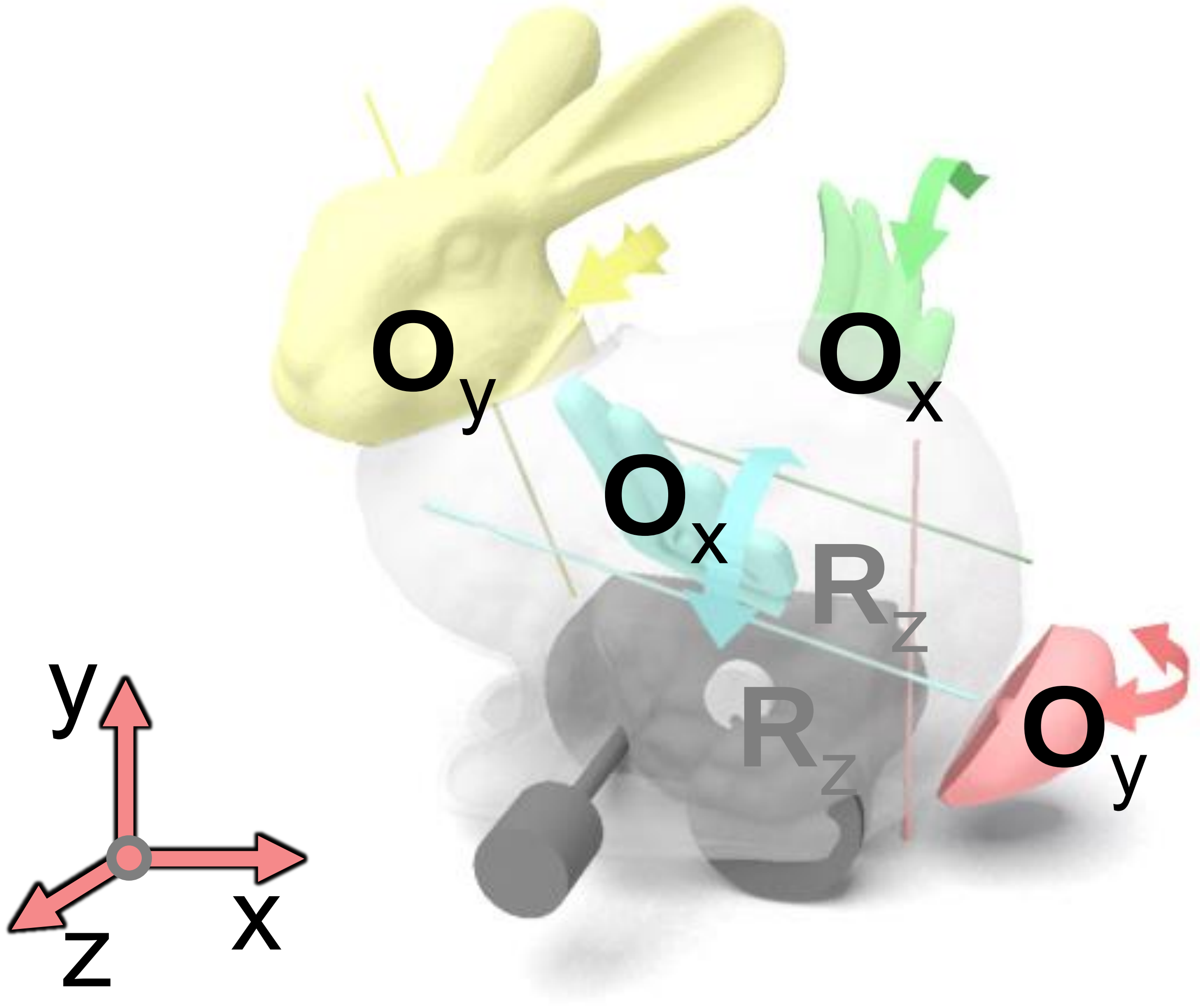



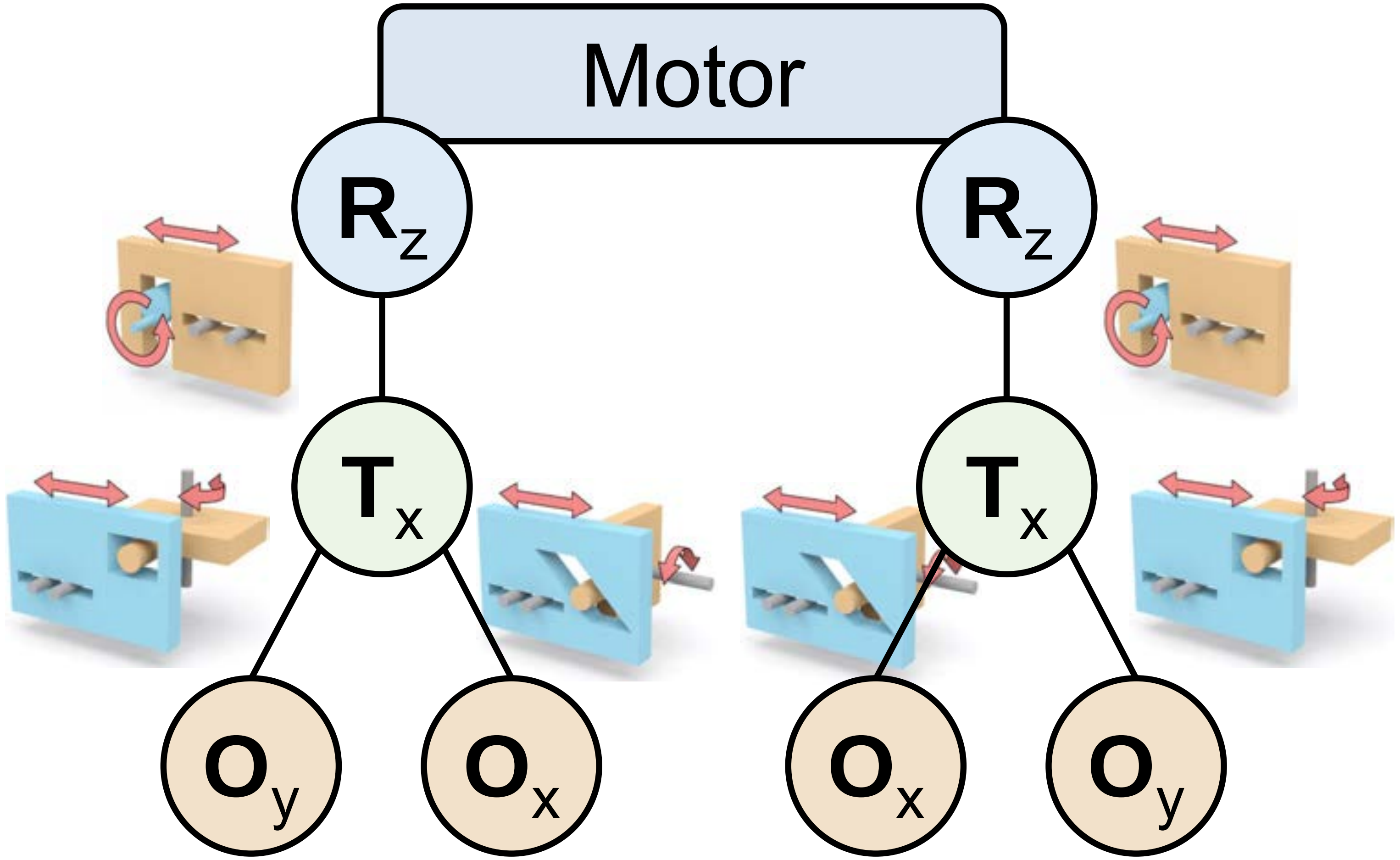
# Conceptual Design: **Construct the Tree**



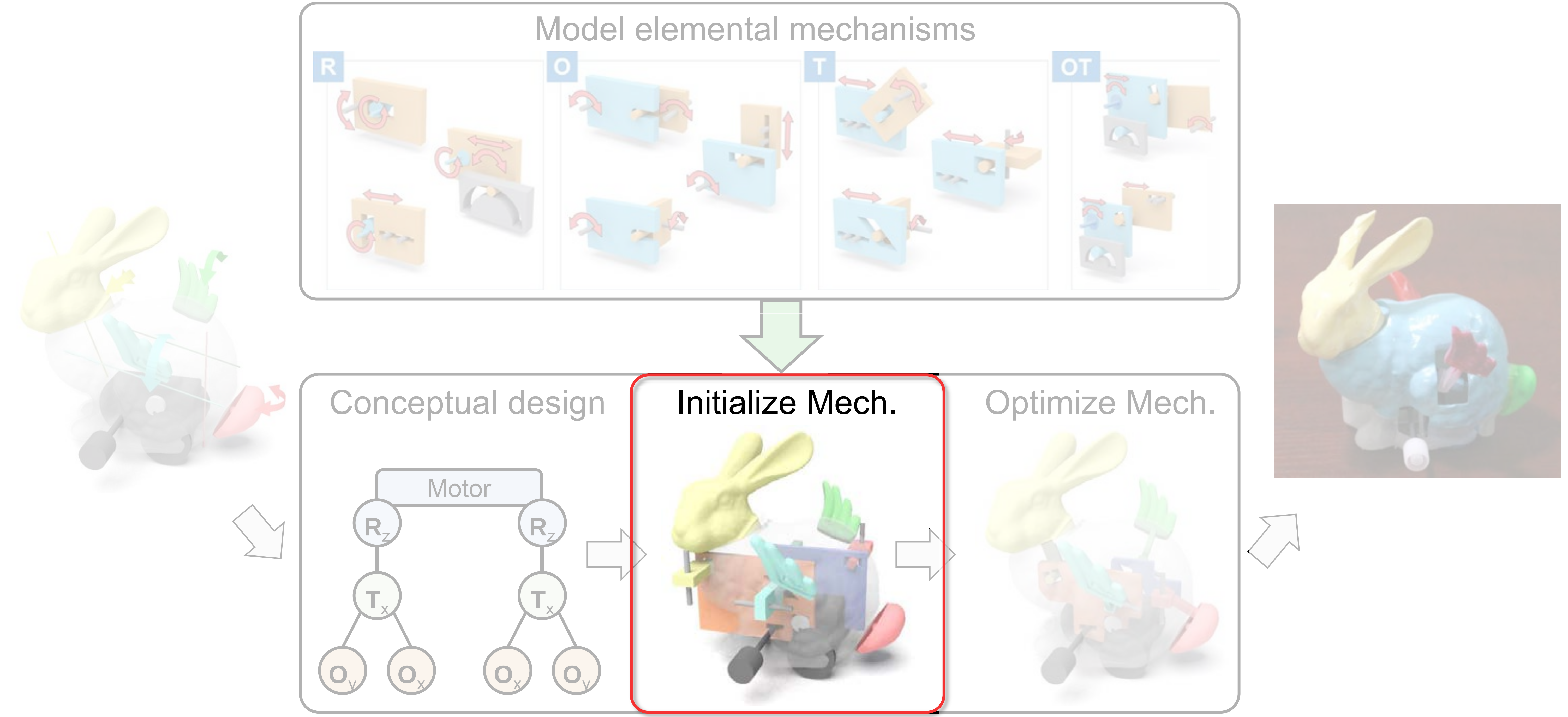



# Conceptual Design: **Construct the Tree**



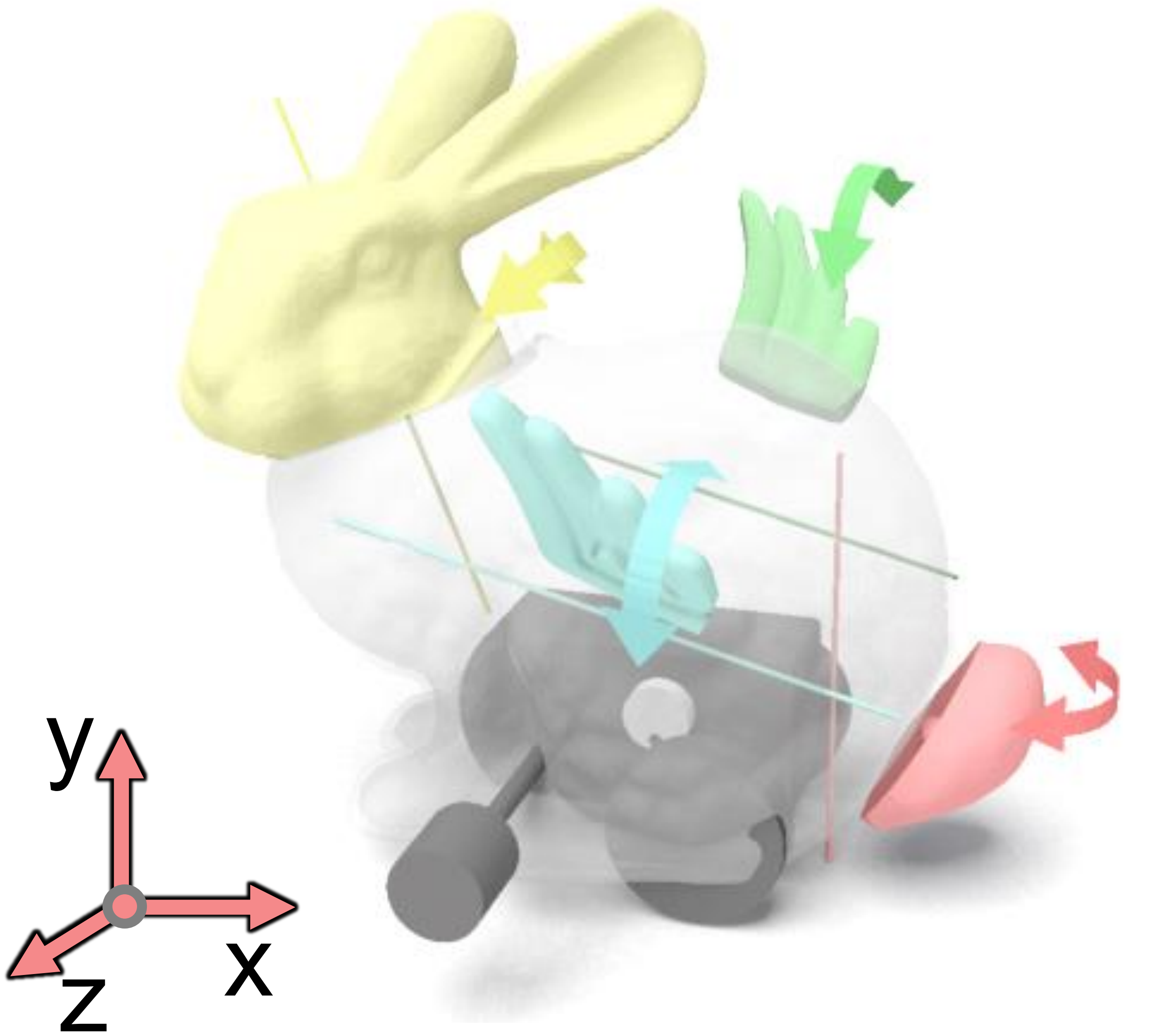



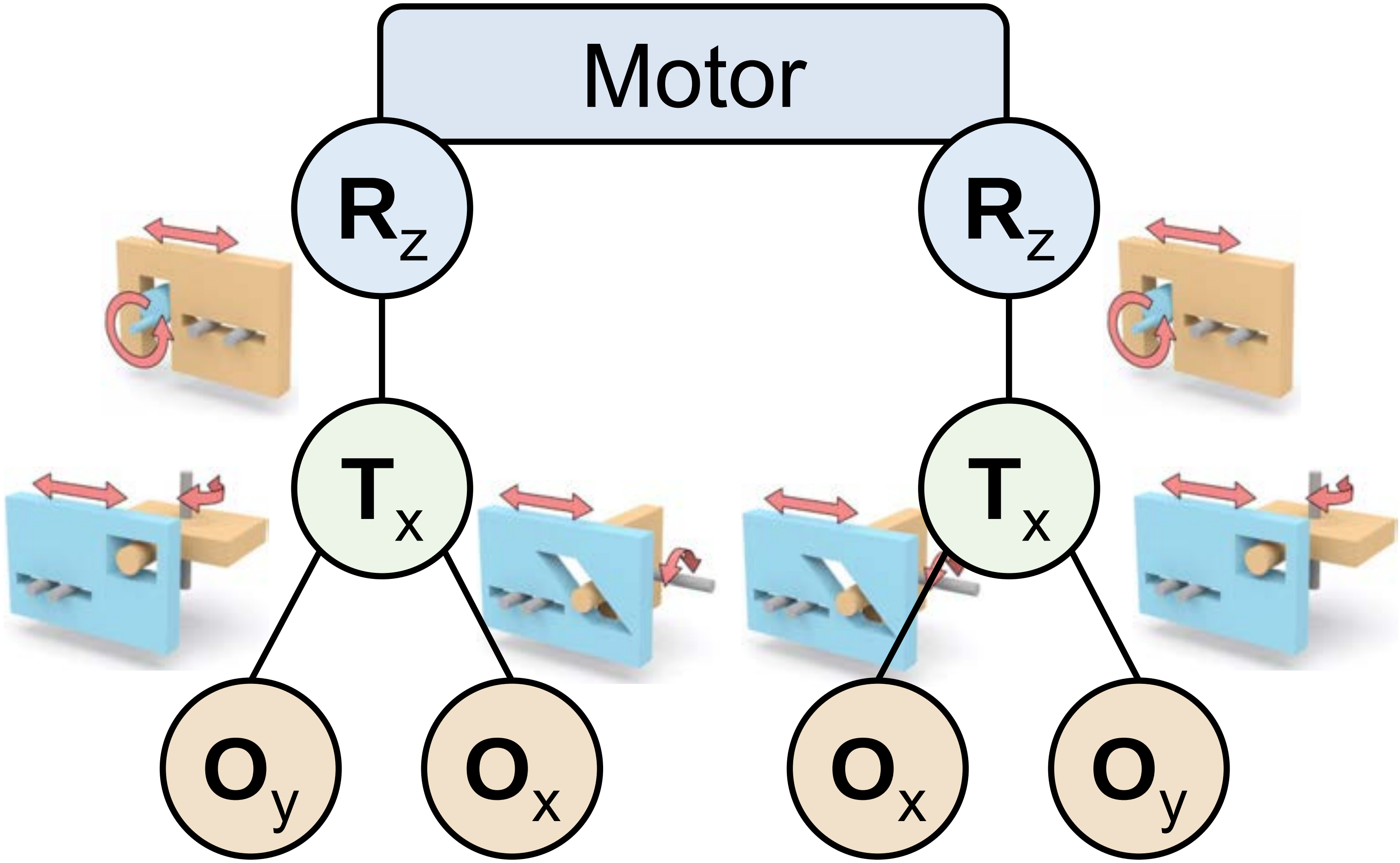
# Overview



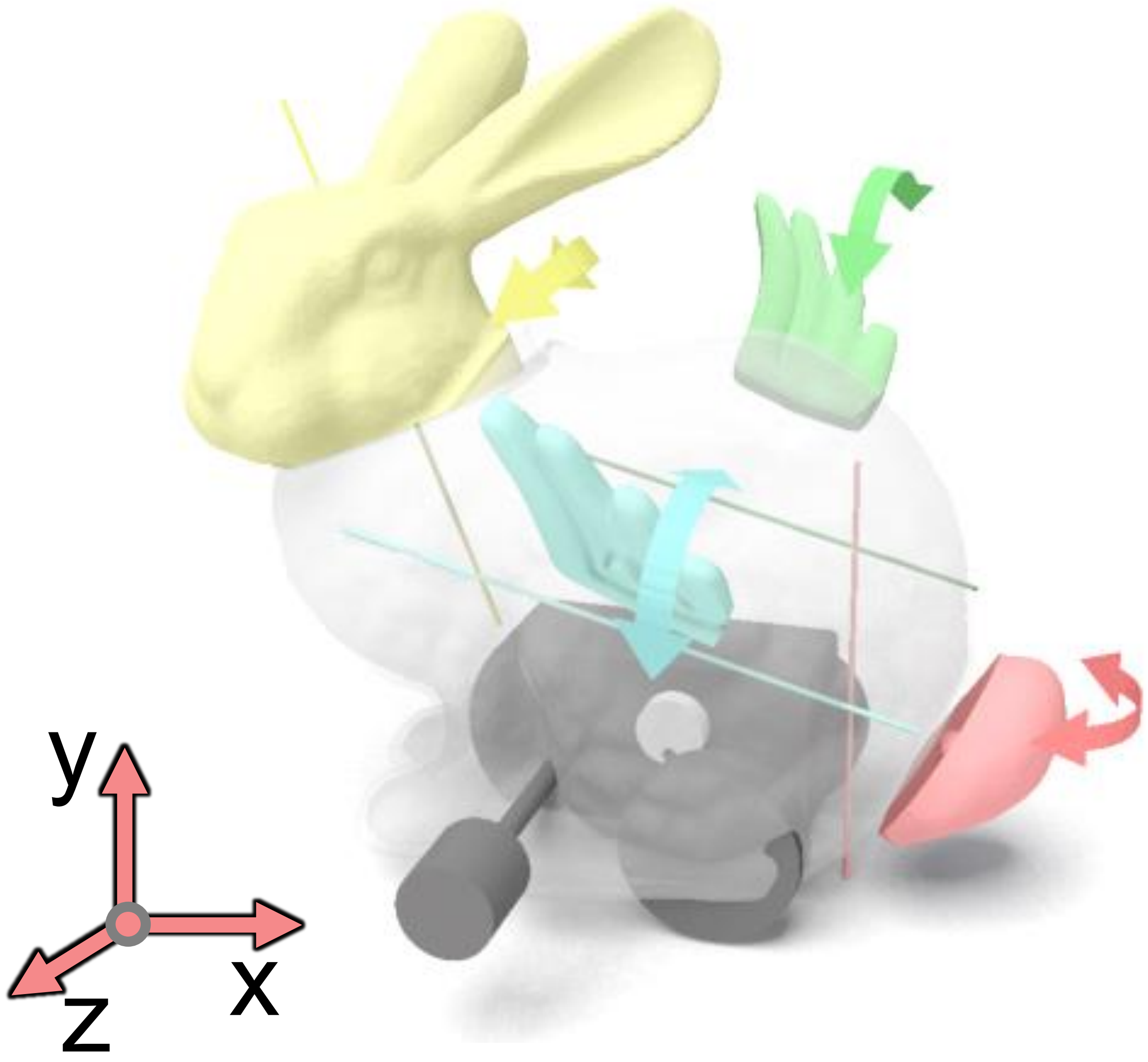
Model elemental mechanisms
R
O
T
OT
Conceptual design
Motor
Initialize Mech.
Optimize Mech.

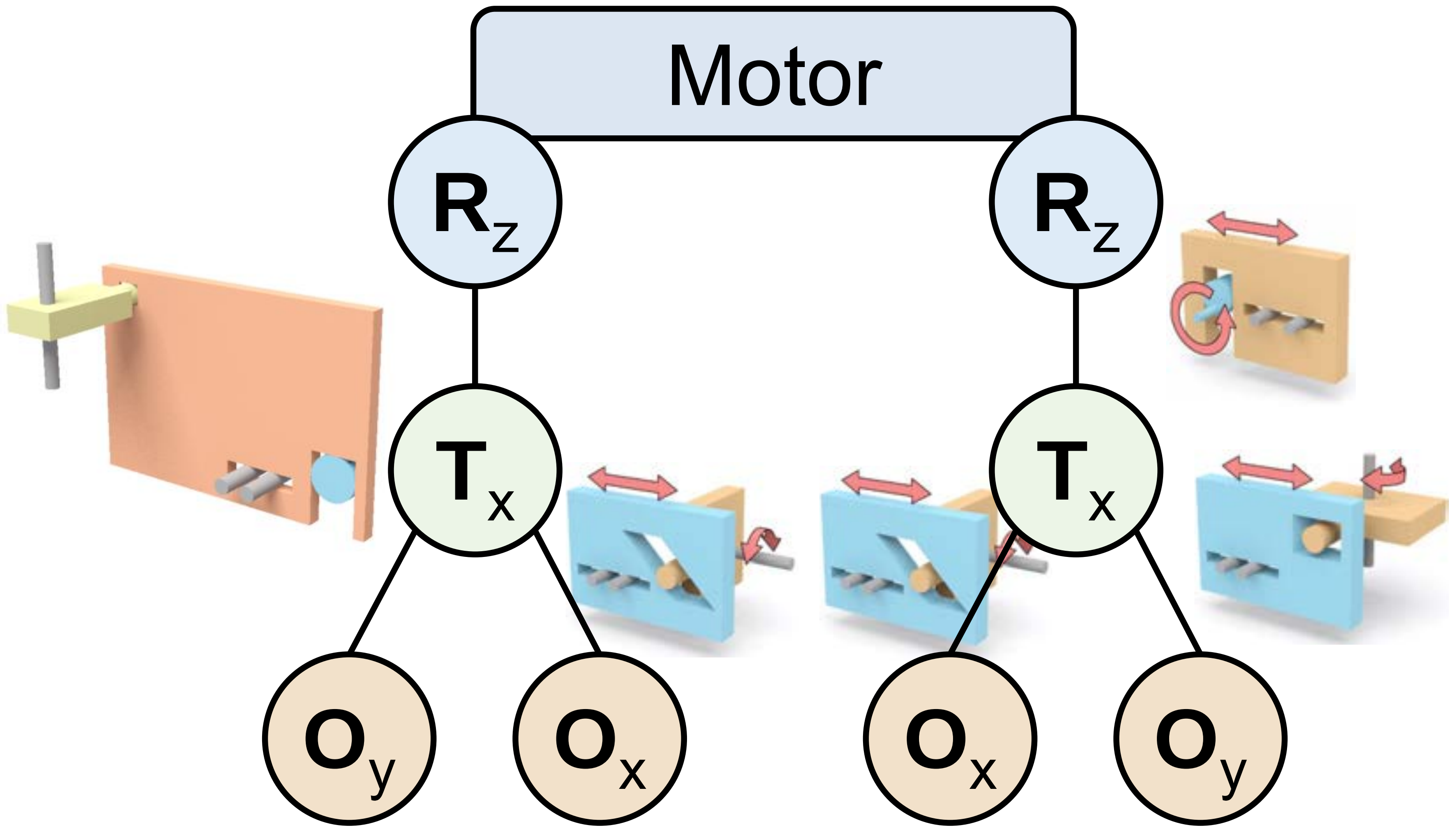
# Initialize Mechanism



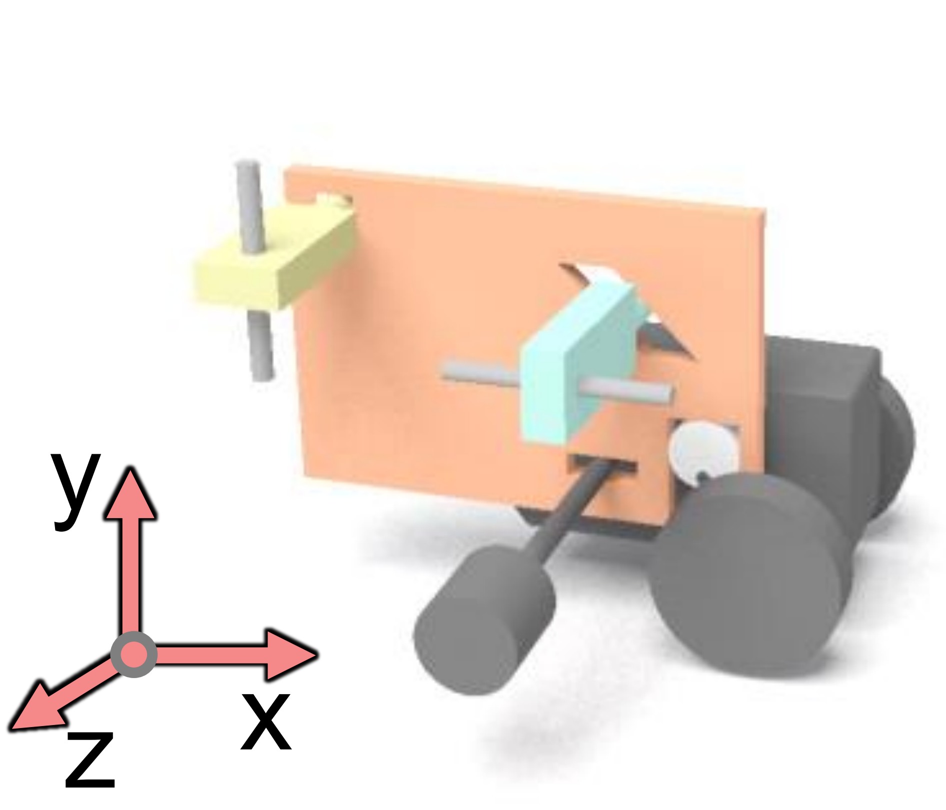



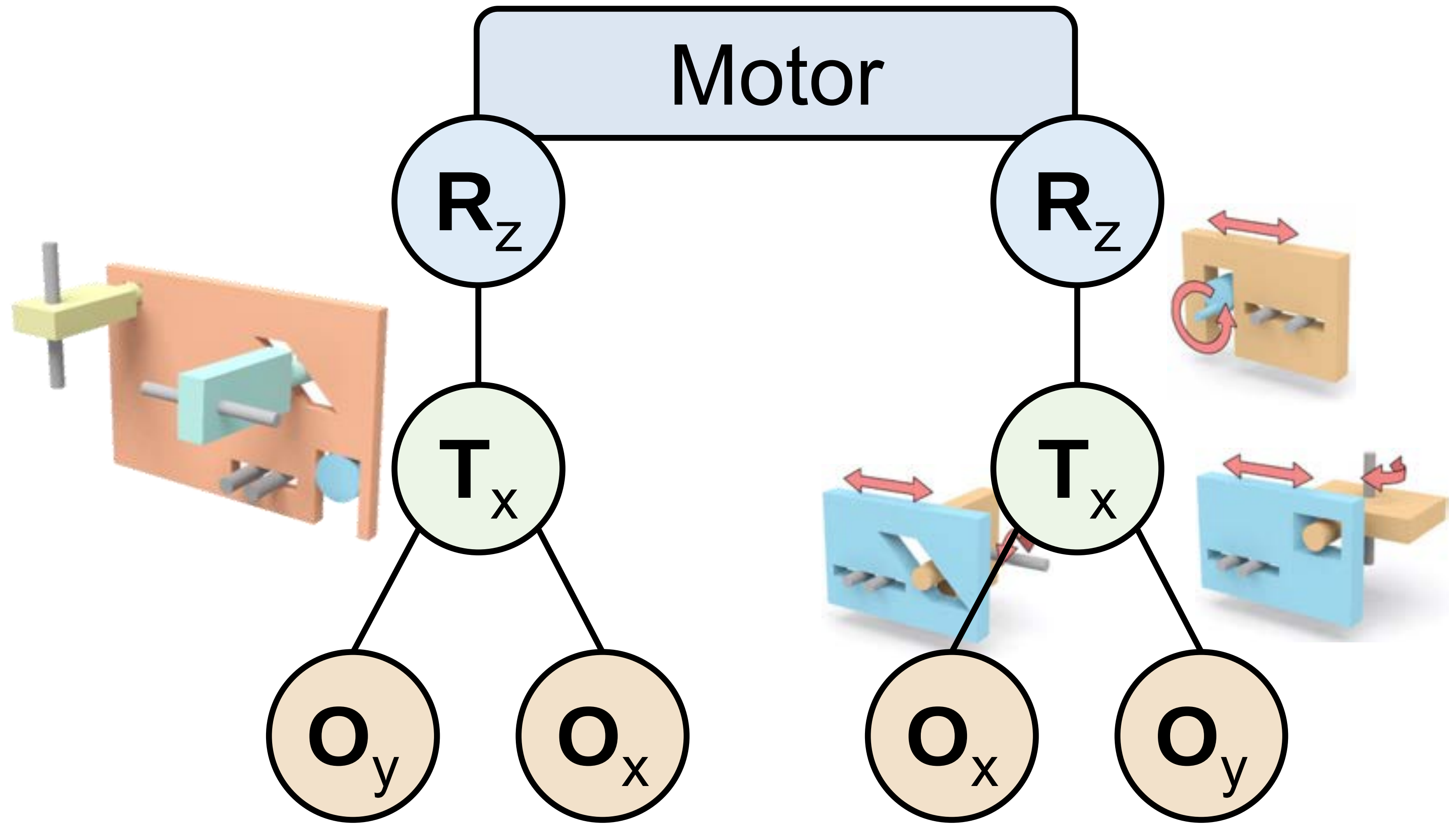
# Initialize Mechanism



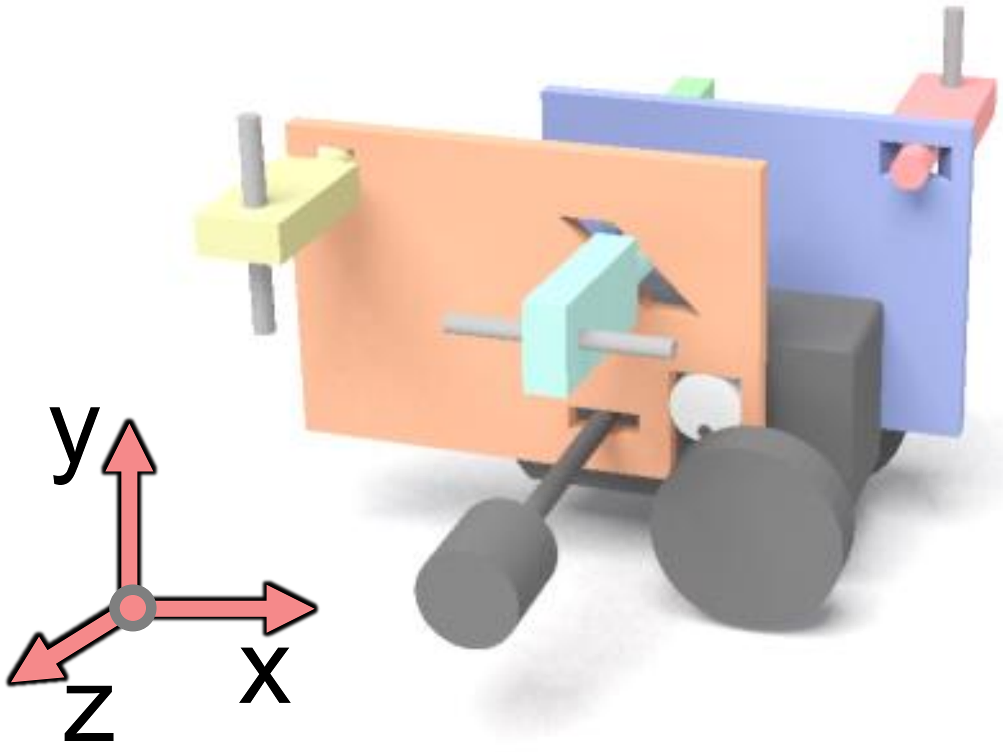



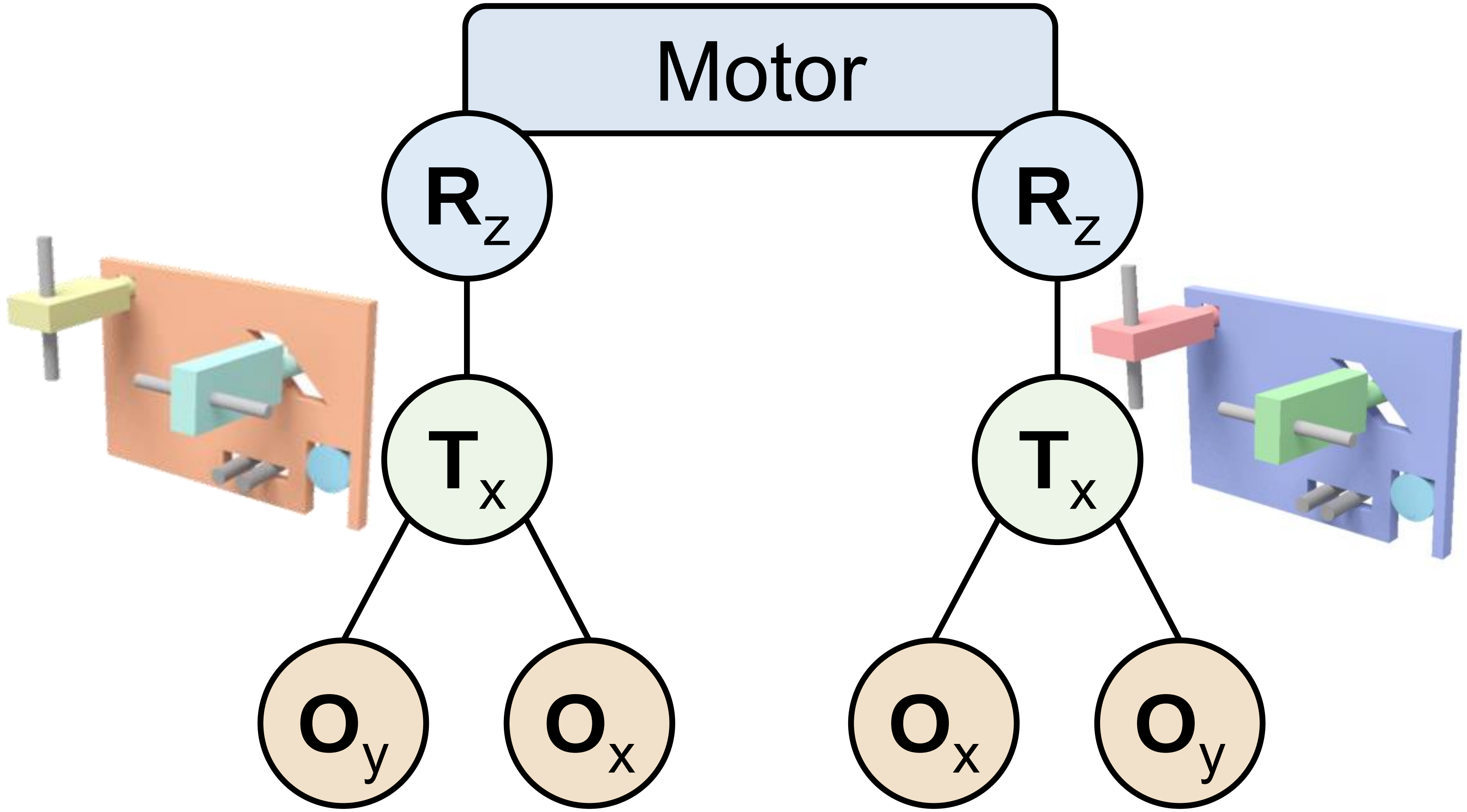
# Initialize Mechanism



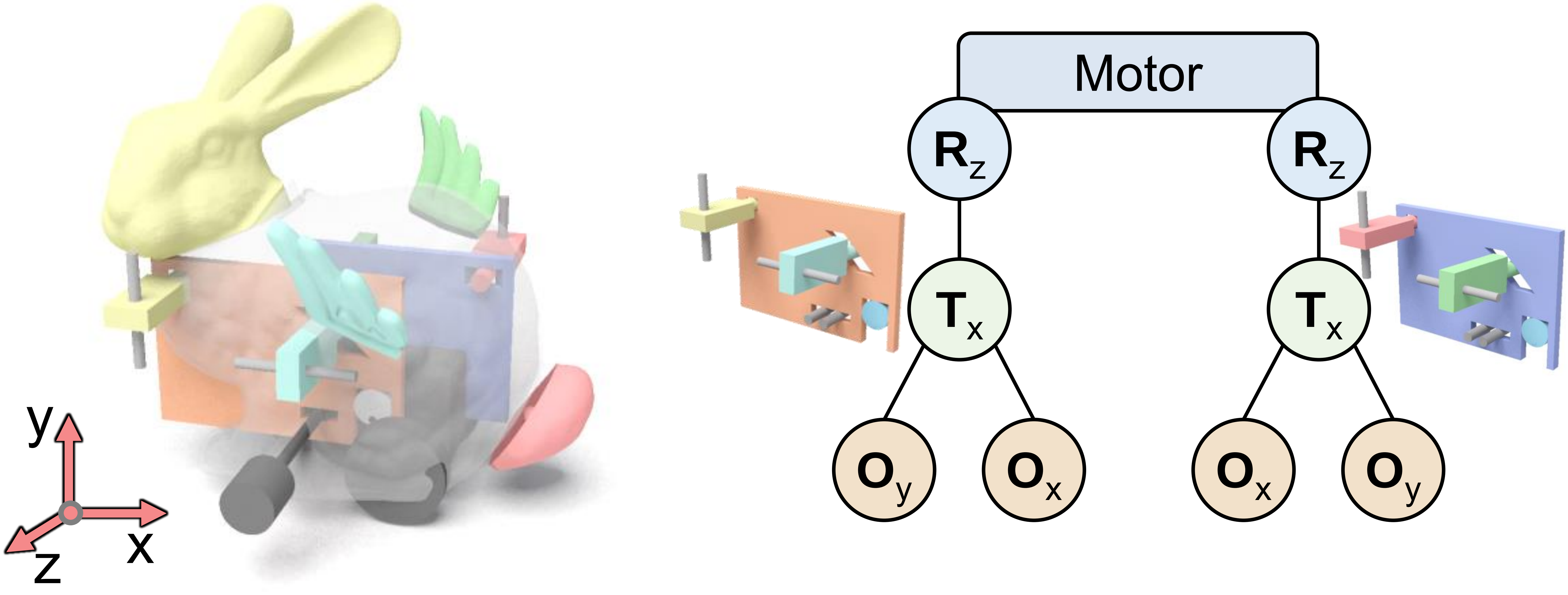



# Initialize Mechanism

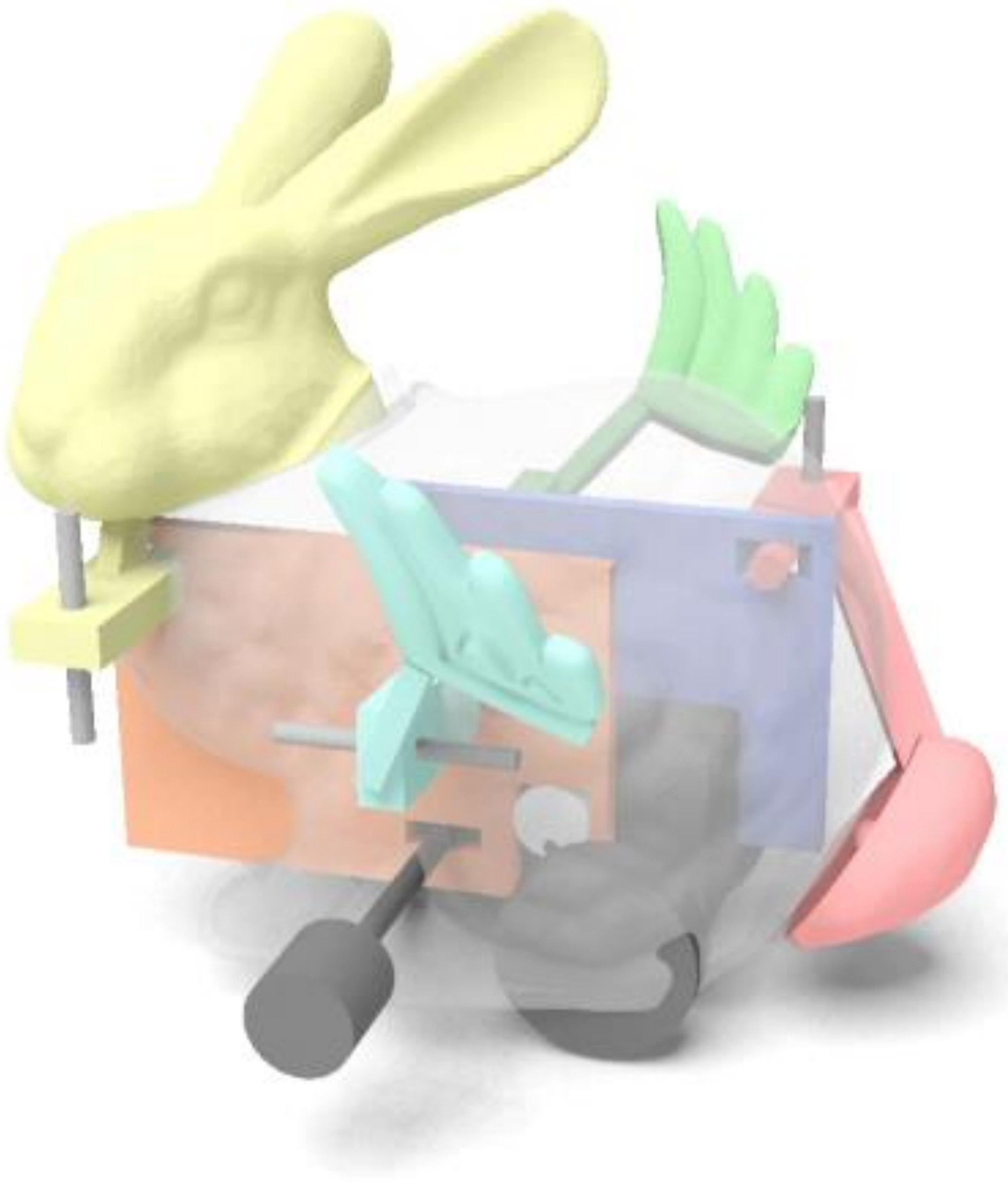


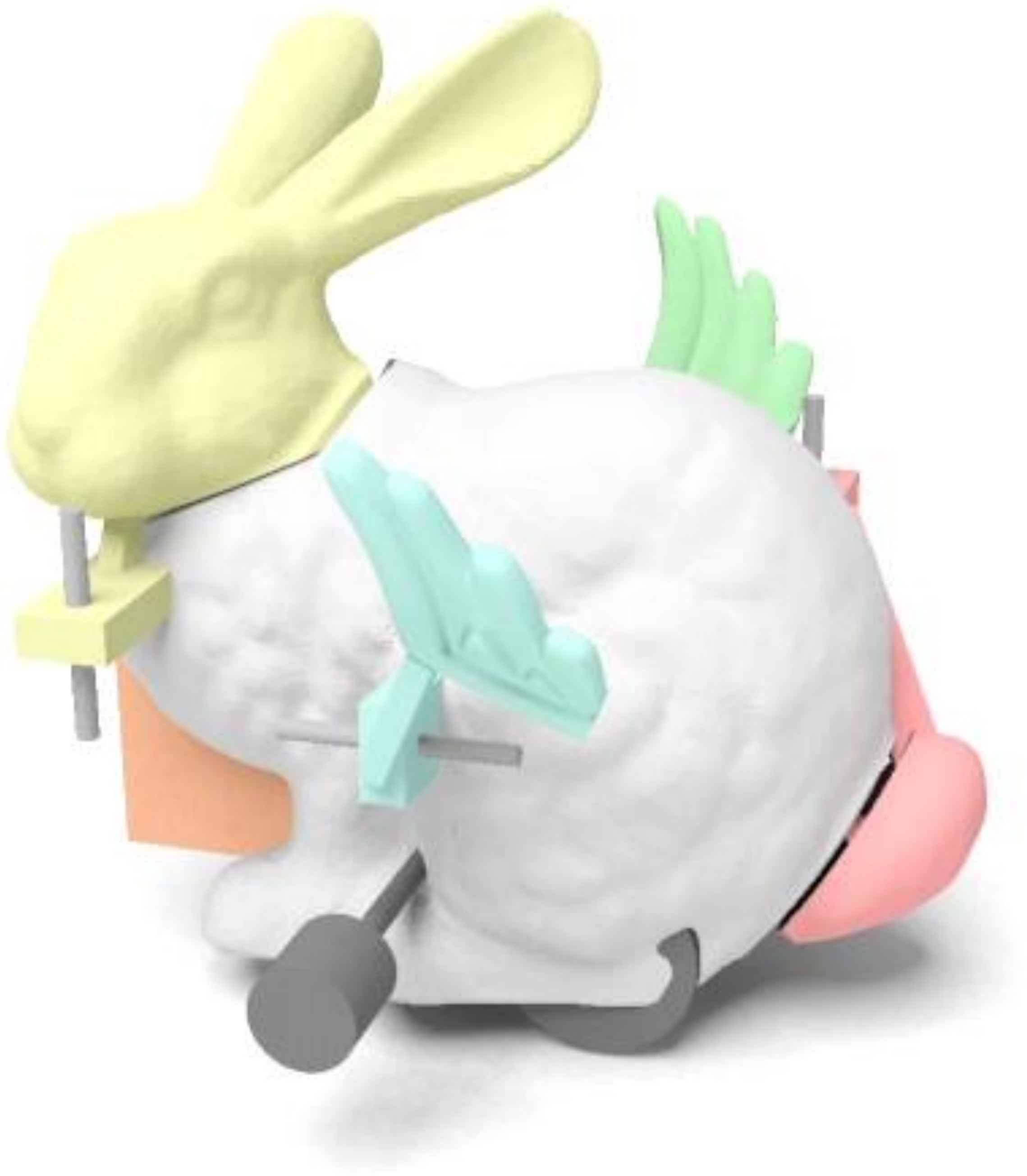
# Initialize Mechanism



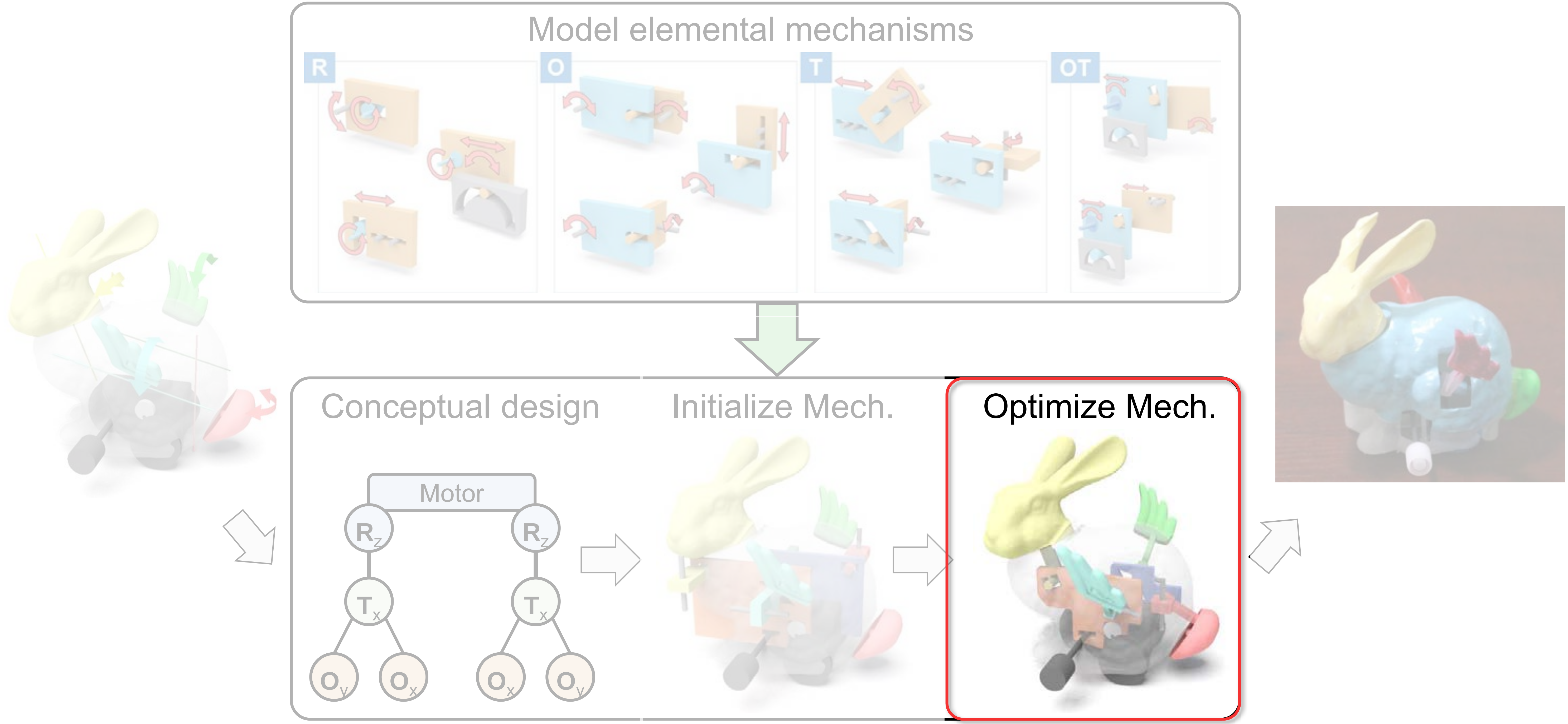
# Initialize Mechanism



# Initialize Mechanism



# Overview



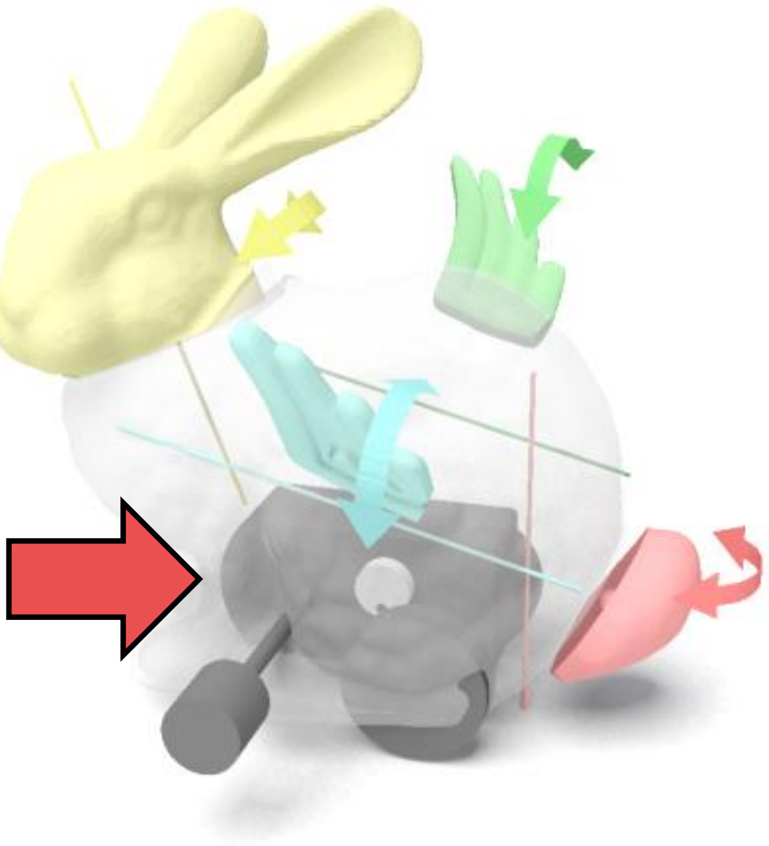
Model elemental mechanisms
R
O
T
OT
Conceptual design
Initialize Mech.
Optimize Mech.
Motor
$R_z$
$R_z$
$T_x$
$T_x$
$O_y$
$O_x$
$O_x$
$O_y$

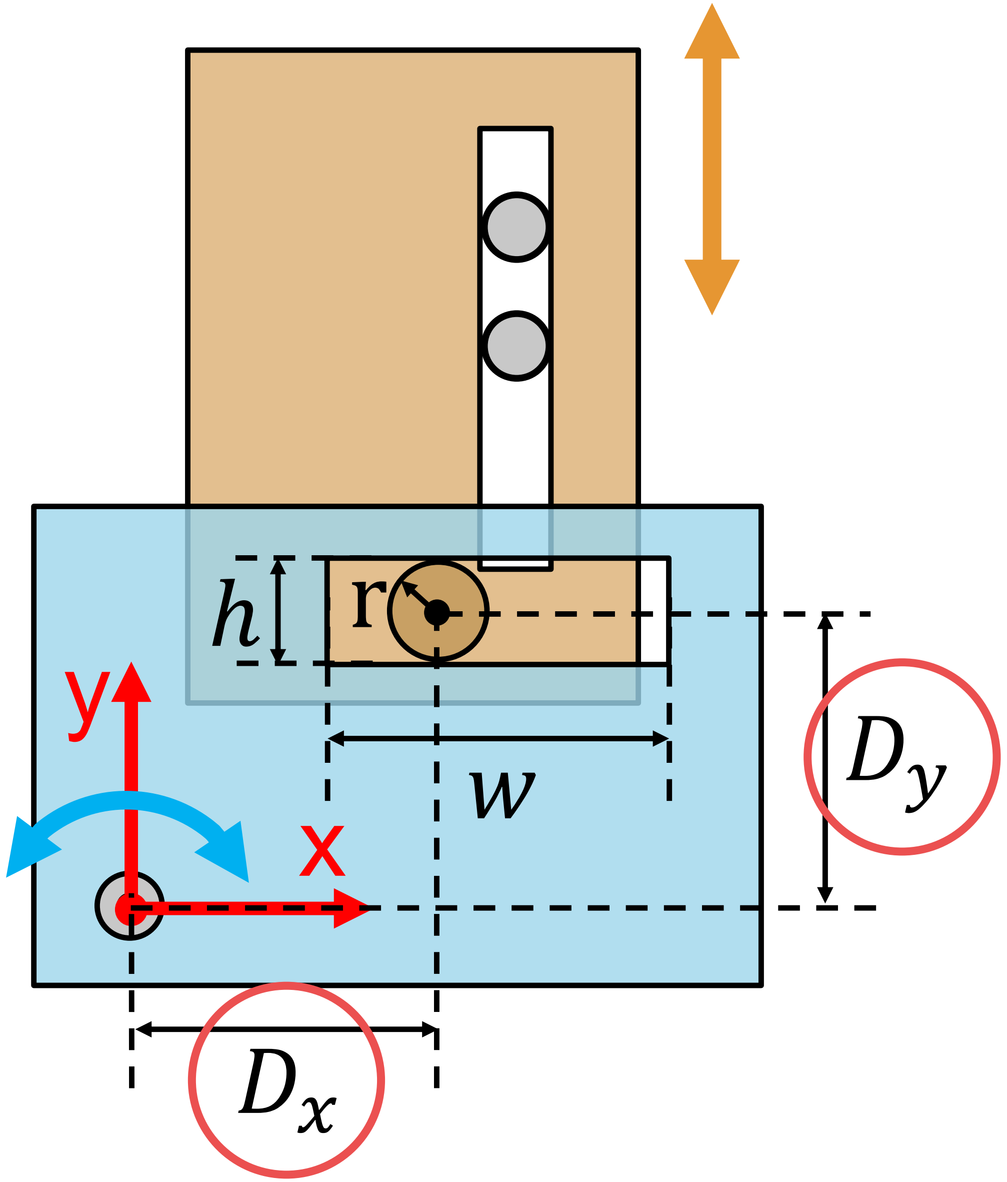
# Optimize Mechanism: **Search Space**

Parameters that define a configuration of the wind-up mechanism

1) motor's pose in the toy space

2) joint layout parameters





# Optimize Mechanism: **Formulation**

Minimize

$$F(g) = \omega_1 F_1(g)$$ total volume of the parts

lightweight

$$+ \omega_2 F_2(g)$$ the maximum signed distance between joints and the body shell

compact

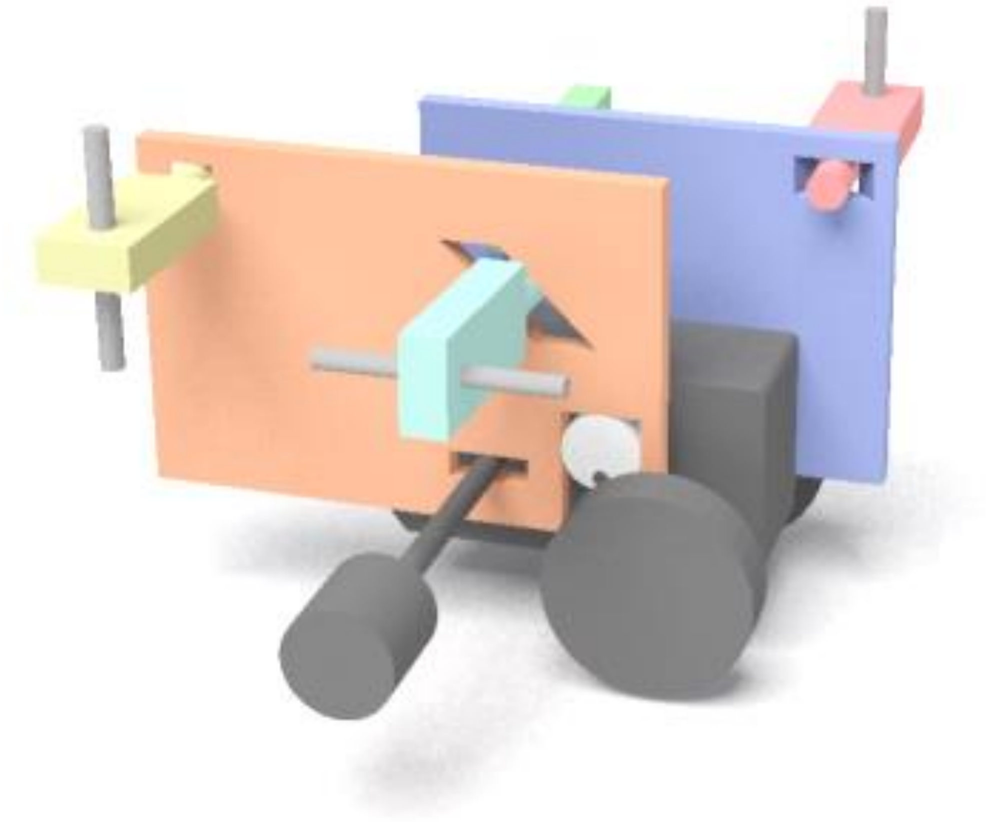
$$+ \omega_3 F_3(g)$$ measure motion similarity (motion axis)

motion

$$+ \omega_4 F_4(g)$$ measure motion similarity (motion range)

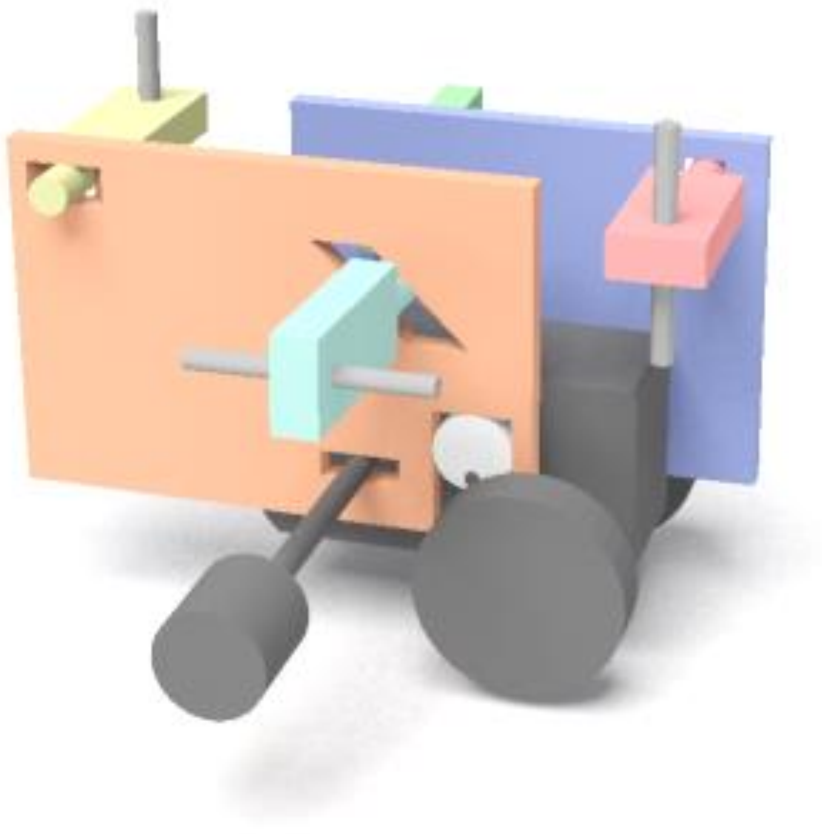
# Coarse Optimization

**Goal**: 1) compact the mechanism;
2) make each end-effector part close to its target location



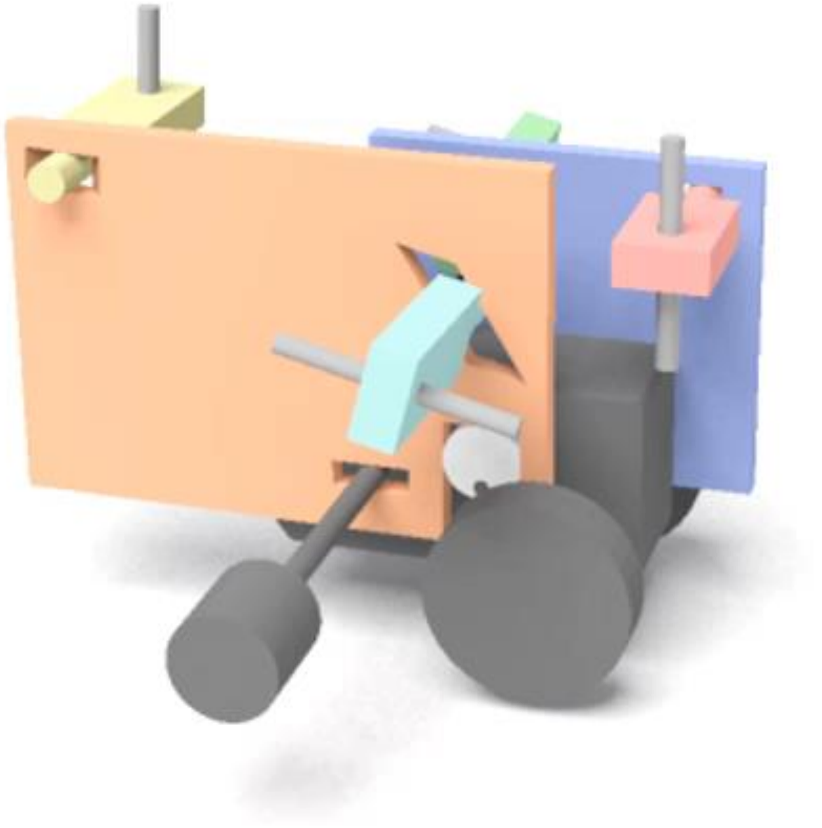
# Coarse Optimization

**Goal**: 1) compact the mechanism;
2) make each end-effector part close to its target location



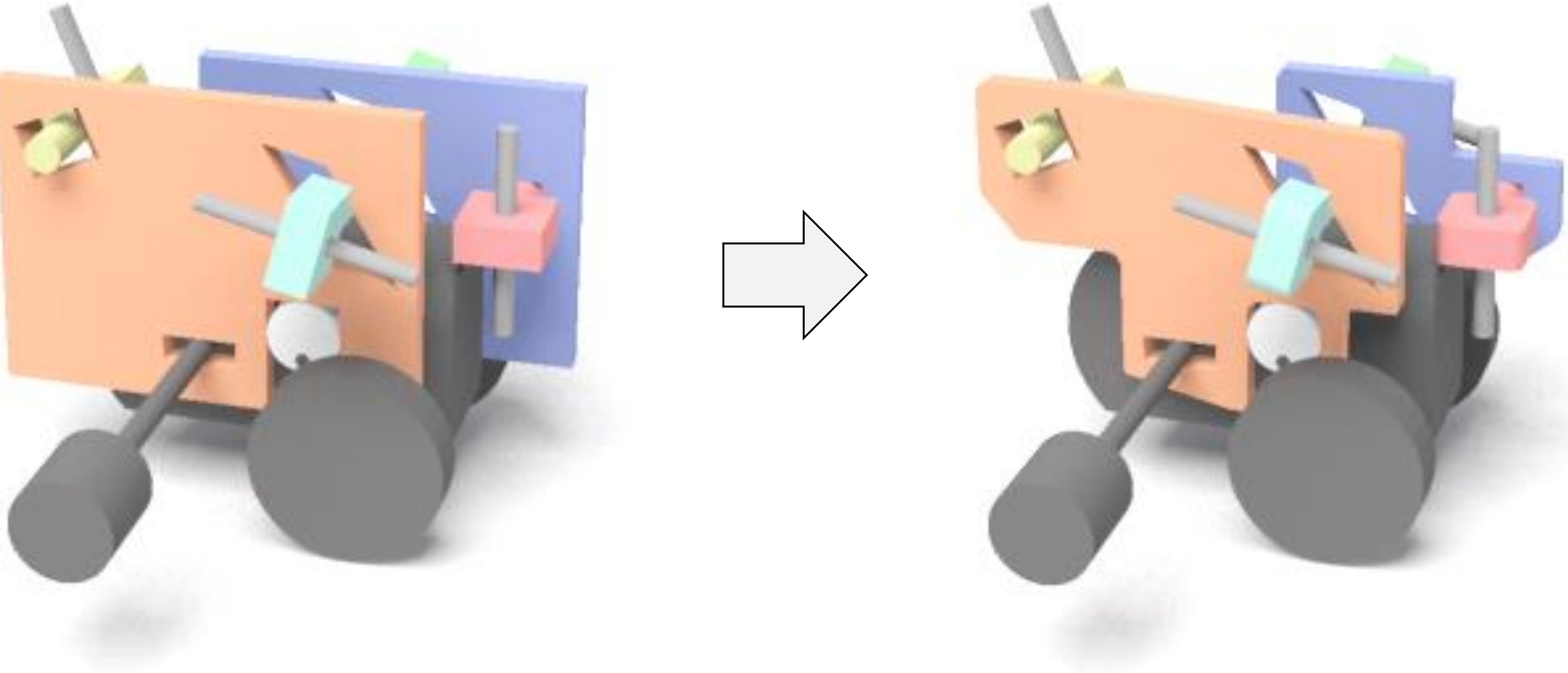
# Fine Optimization

**Goal**: minimize the whole objective function, particularly for better matching user prescribed end effector motions.



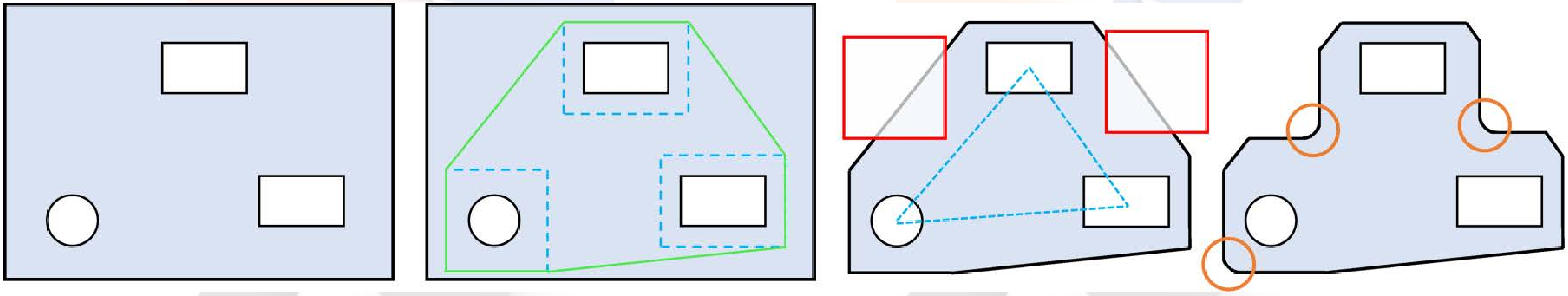
# Fine Optimization

Refine boundary shape of mechanical parts

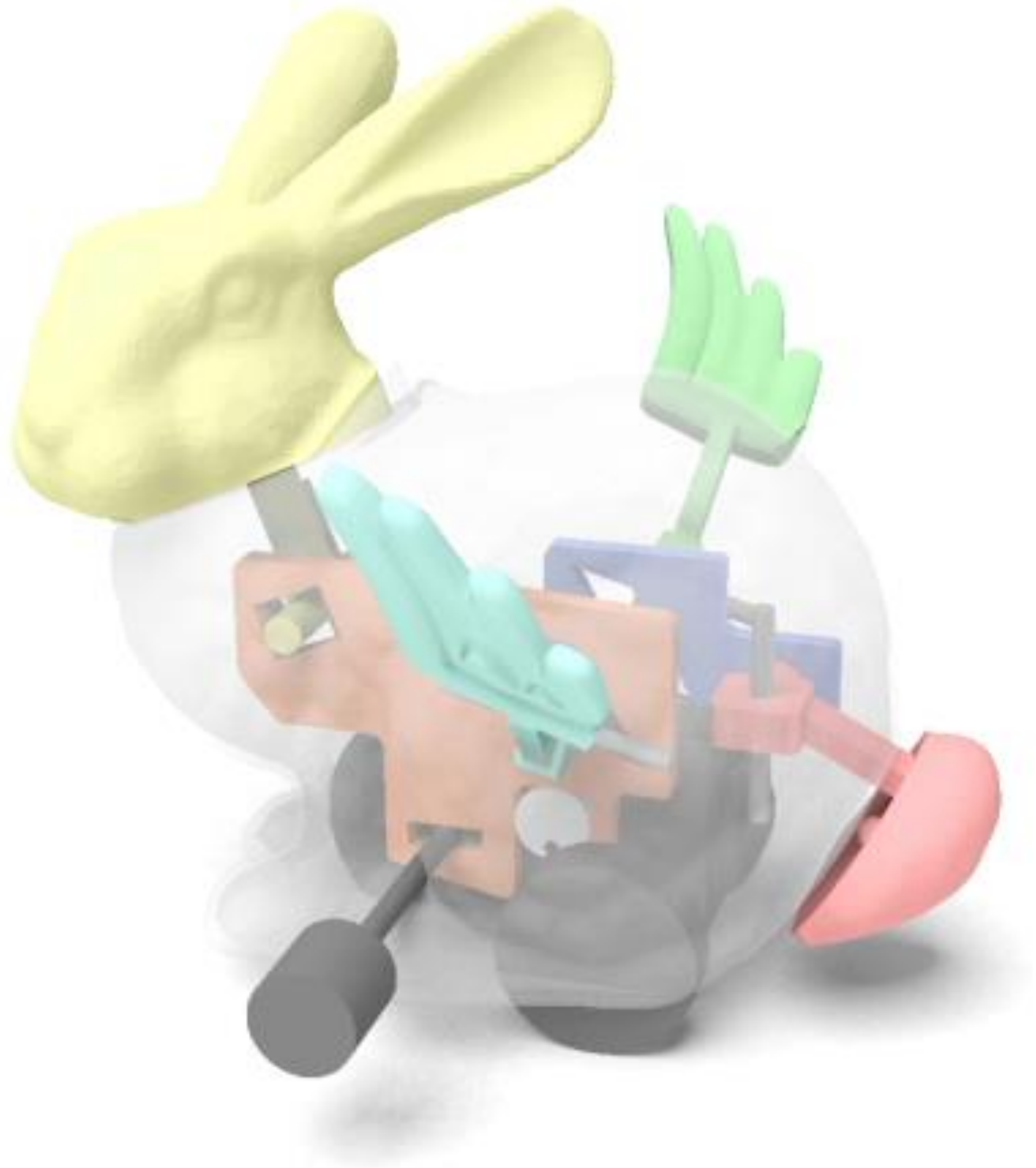


# Fine Optimization

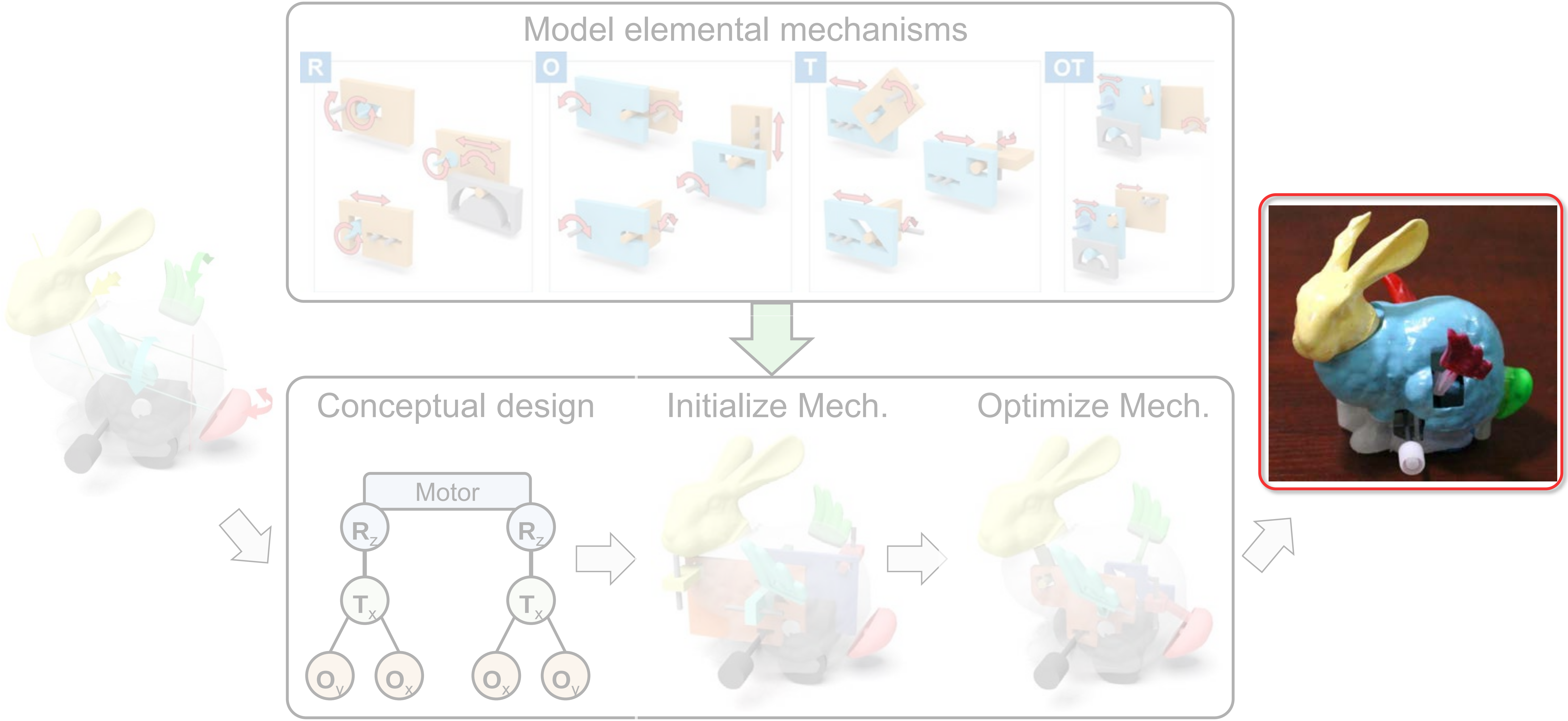
Refine boundary shape of mechanical parts



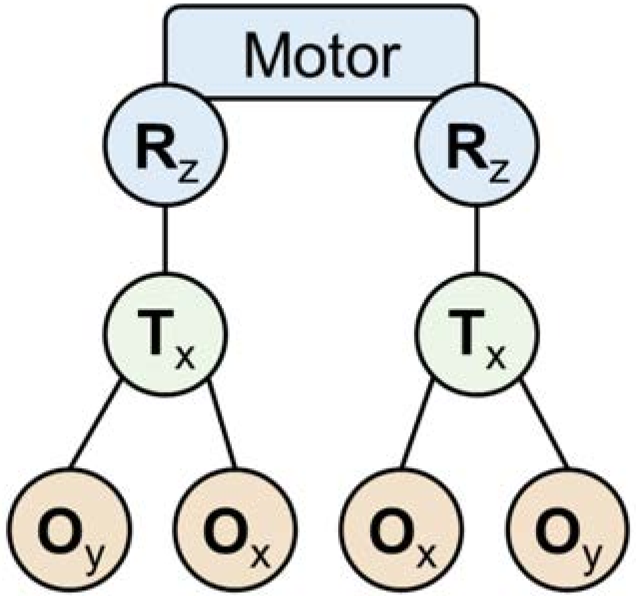
# Final Result



# Results

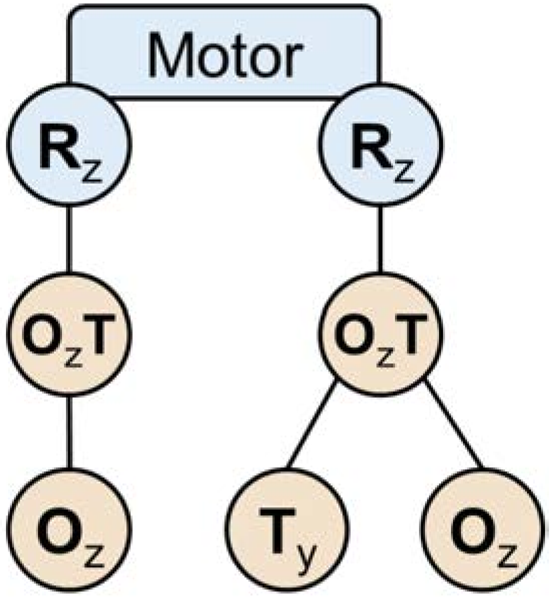

Model elemental mechanisms
R
O
T
OT
Conceptual design
Initialize Mech.
Optimize Mech.
Motor
$R_z$
$R_z$
$T_x$
$T_x$
$O_y$
$O_x$
$O_x$
$O_y$

# **Results:** Bunny



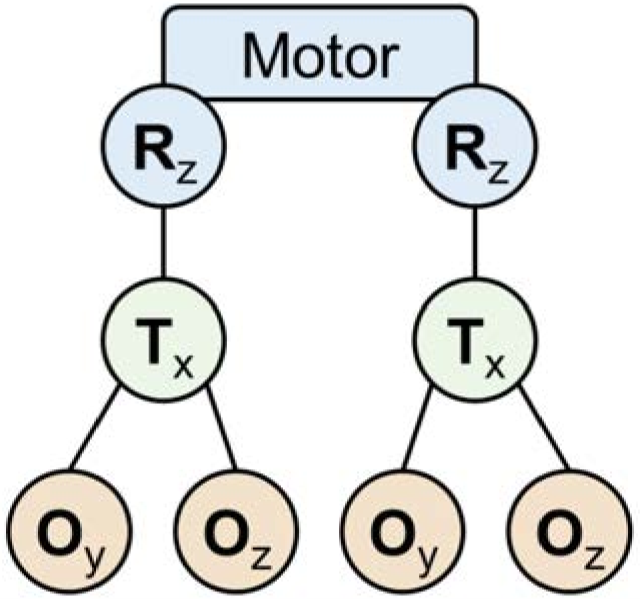

13.0g shell
7.8g motor
9.0g mech.

# **Results:** Horse



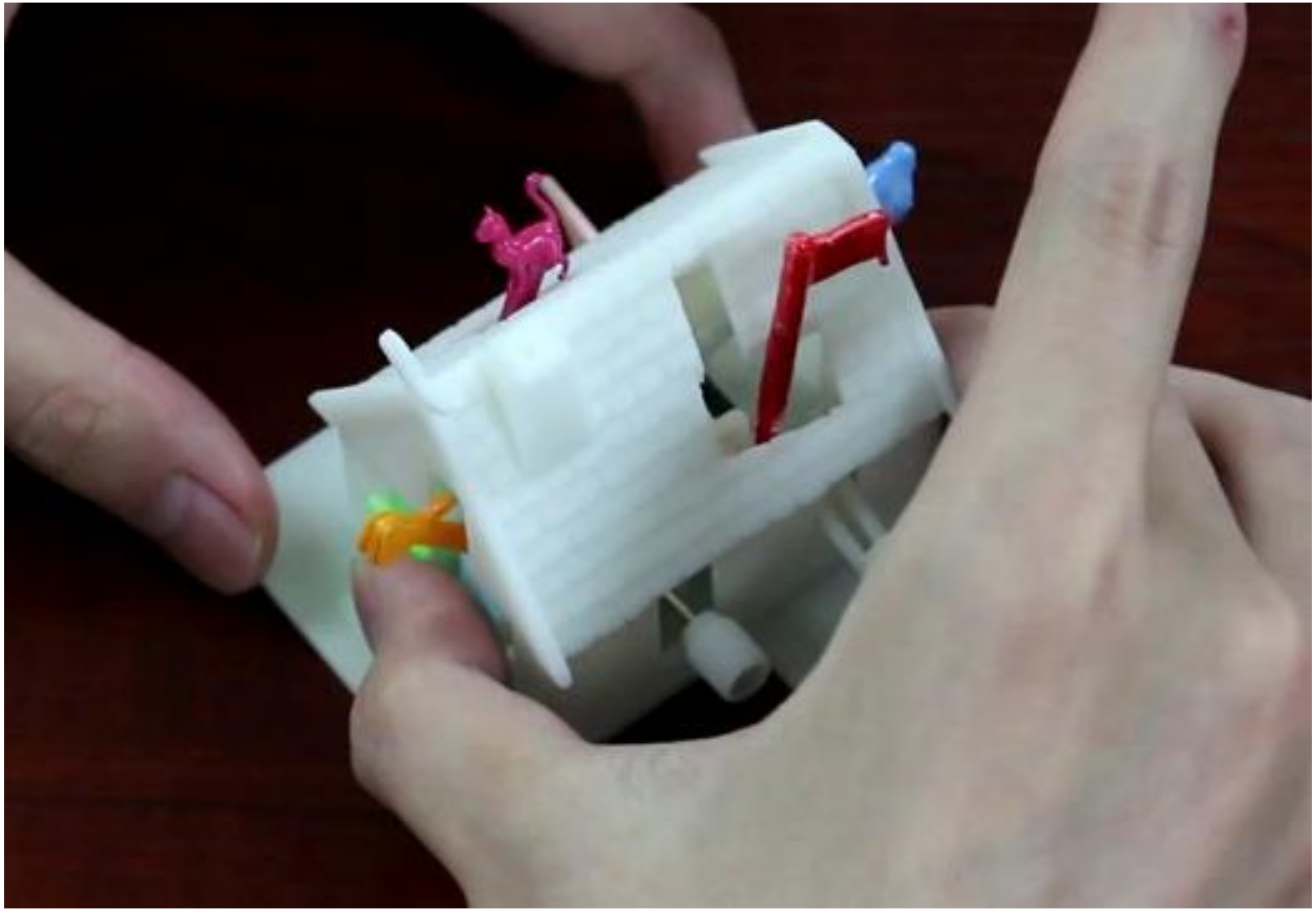

15.3g shell
5.9g motor
26.1g mech.

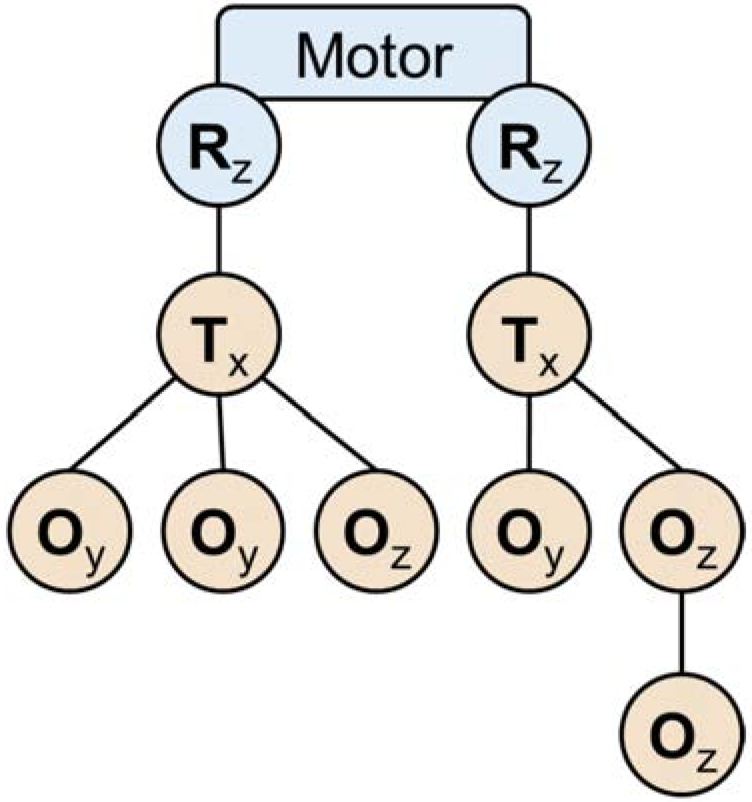
# **Results:** Flower Pot




34.8g shell
5.8g motor
8.0g mech.

# **Results:** House

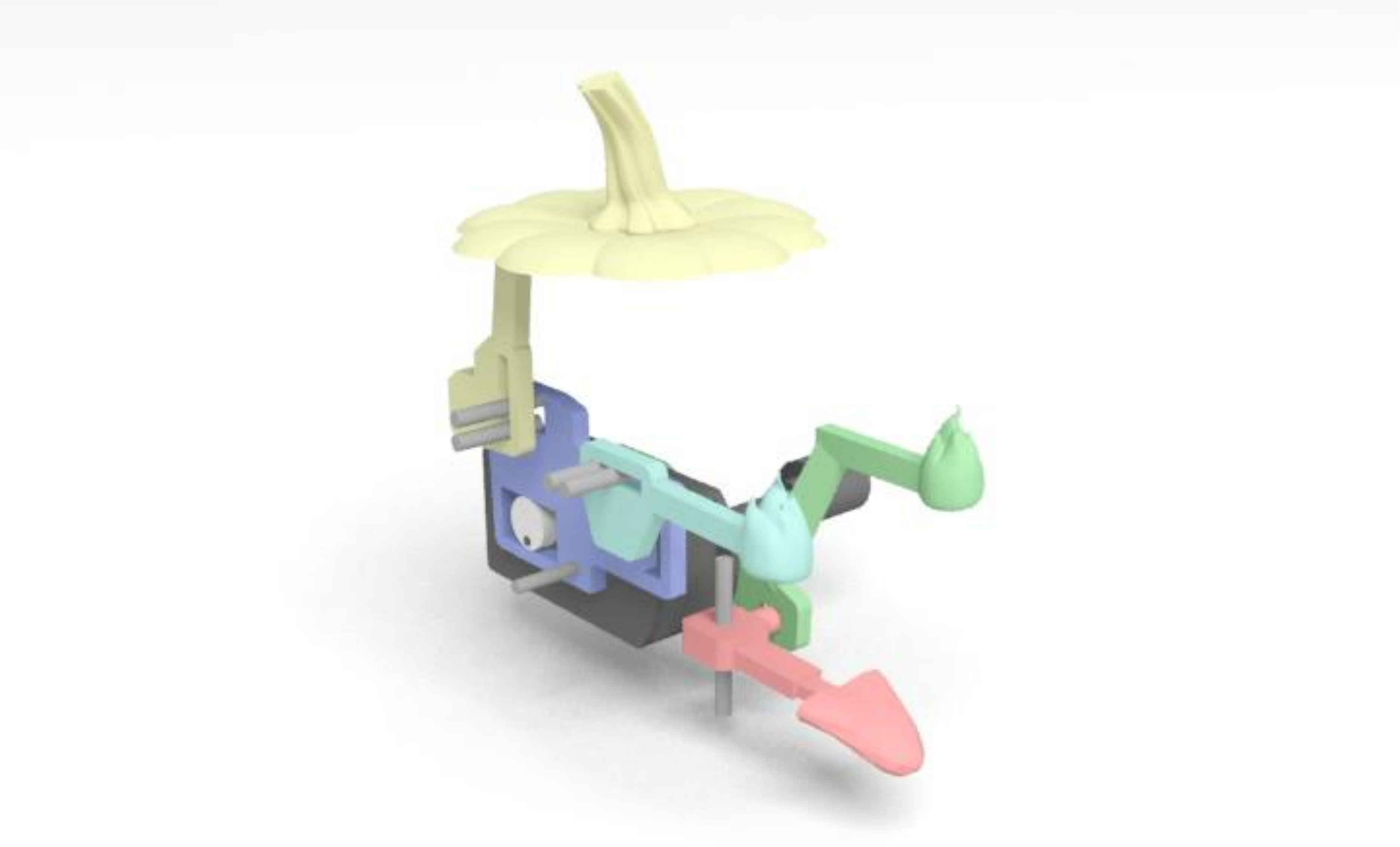


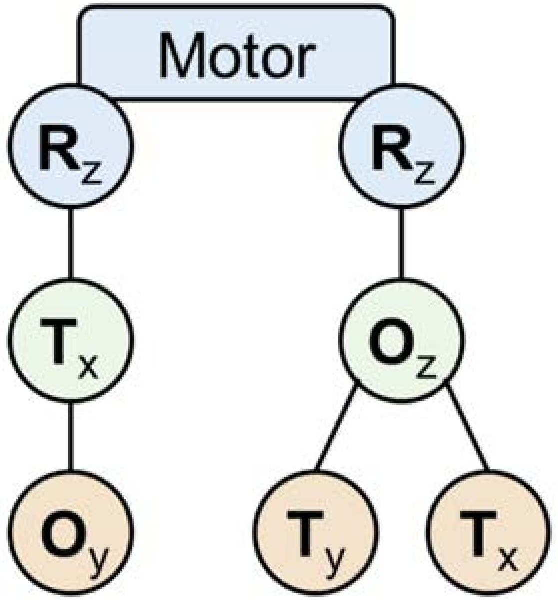



48.9g shell
6.0g motor
6.4g mech.

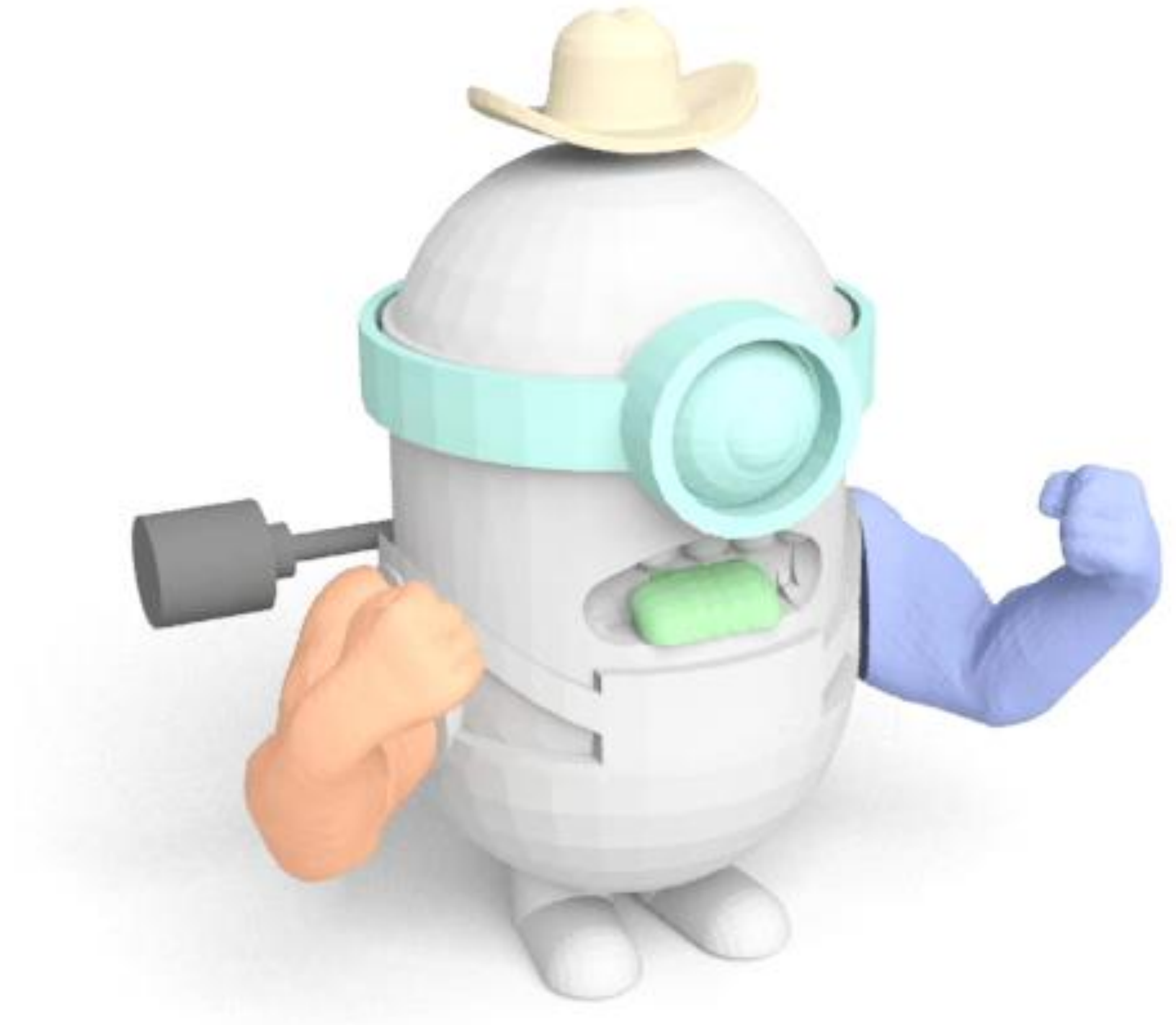
# **Results:** Jack O'Lantern

# **Results:** Jack O'Lantern

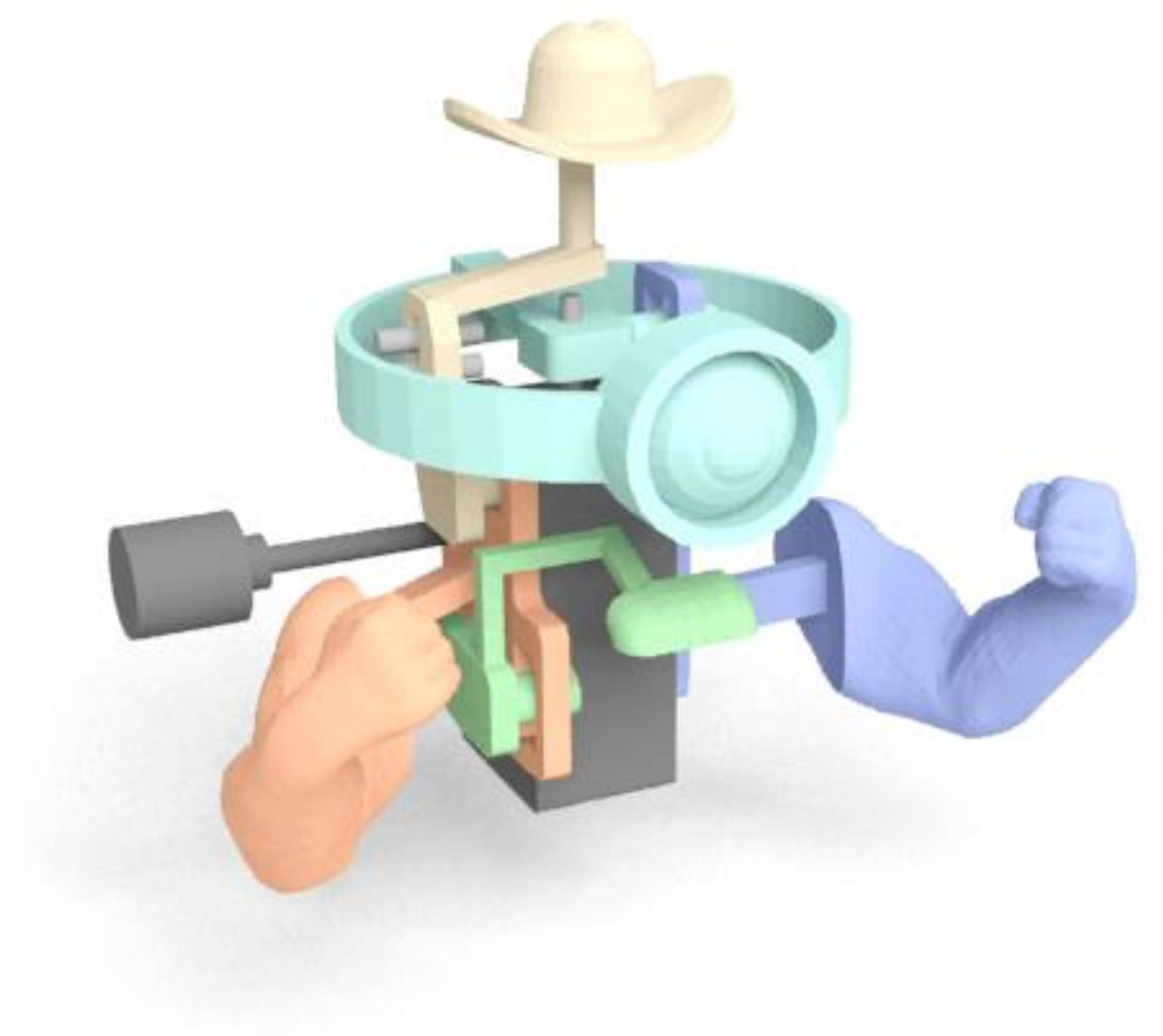


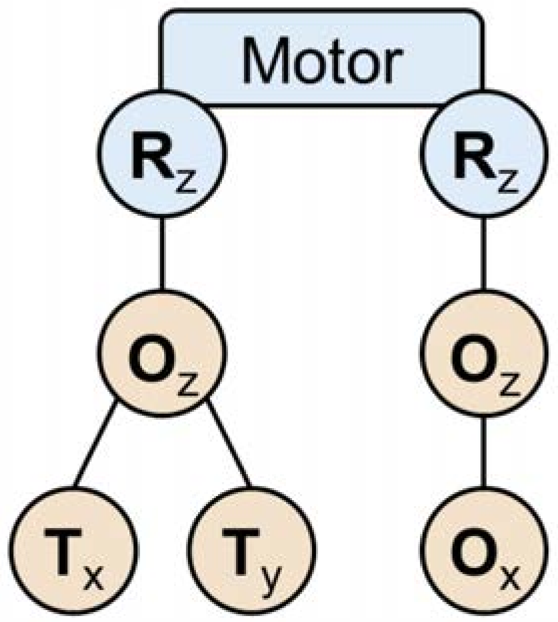


# **Results:** Minion

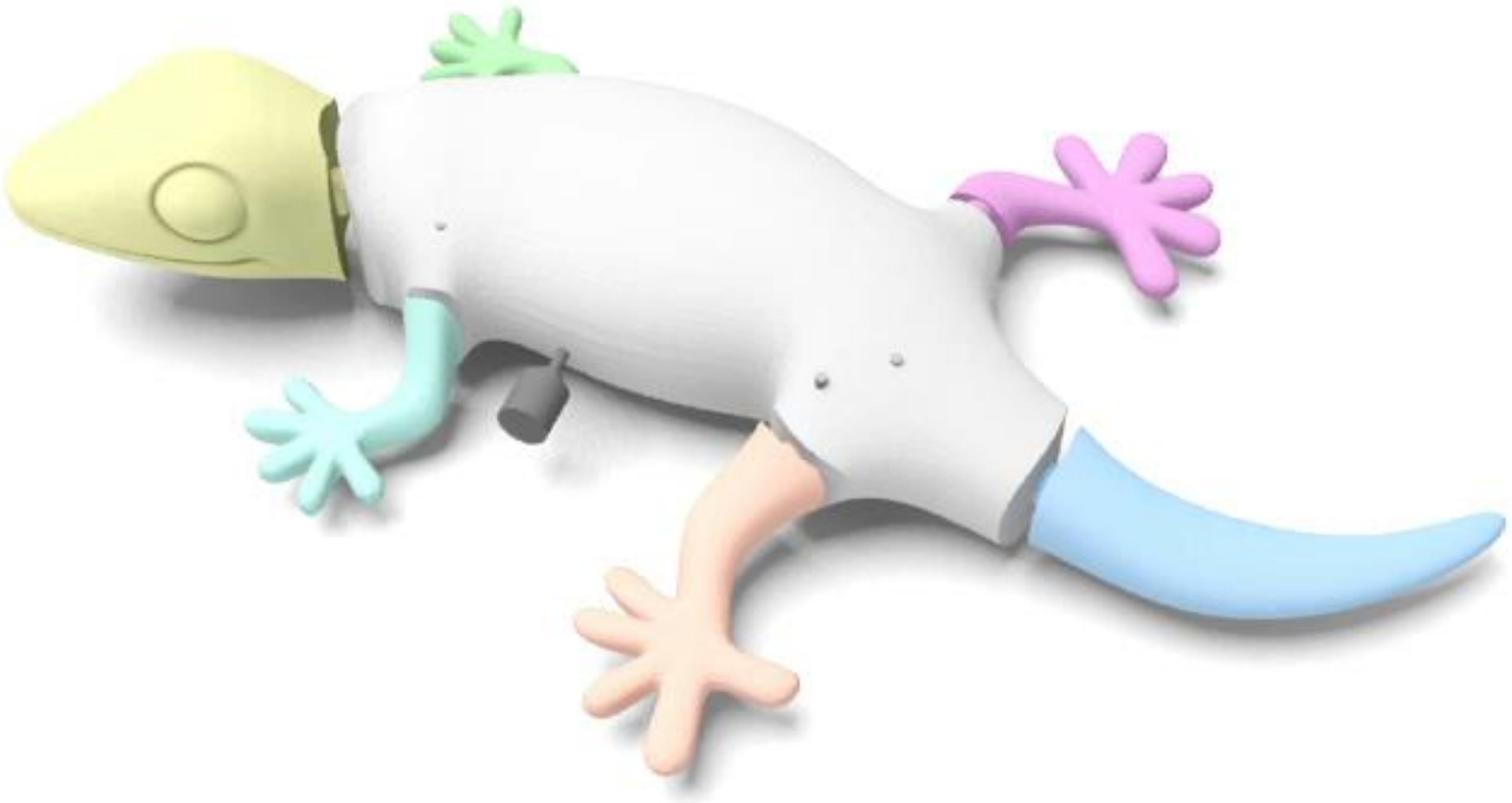


# **Results:** Minion

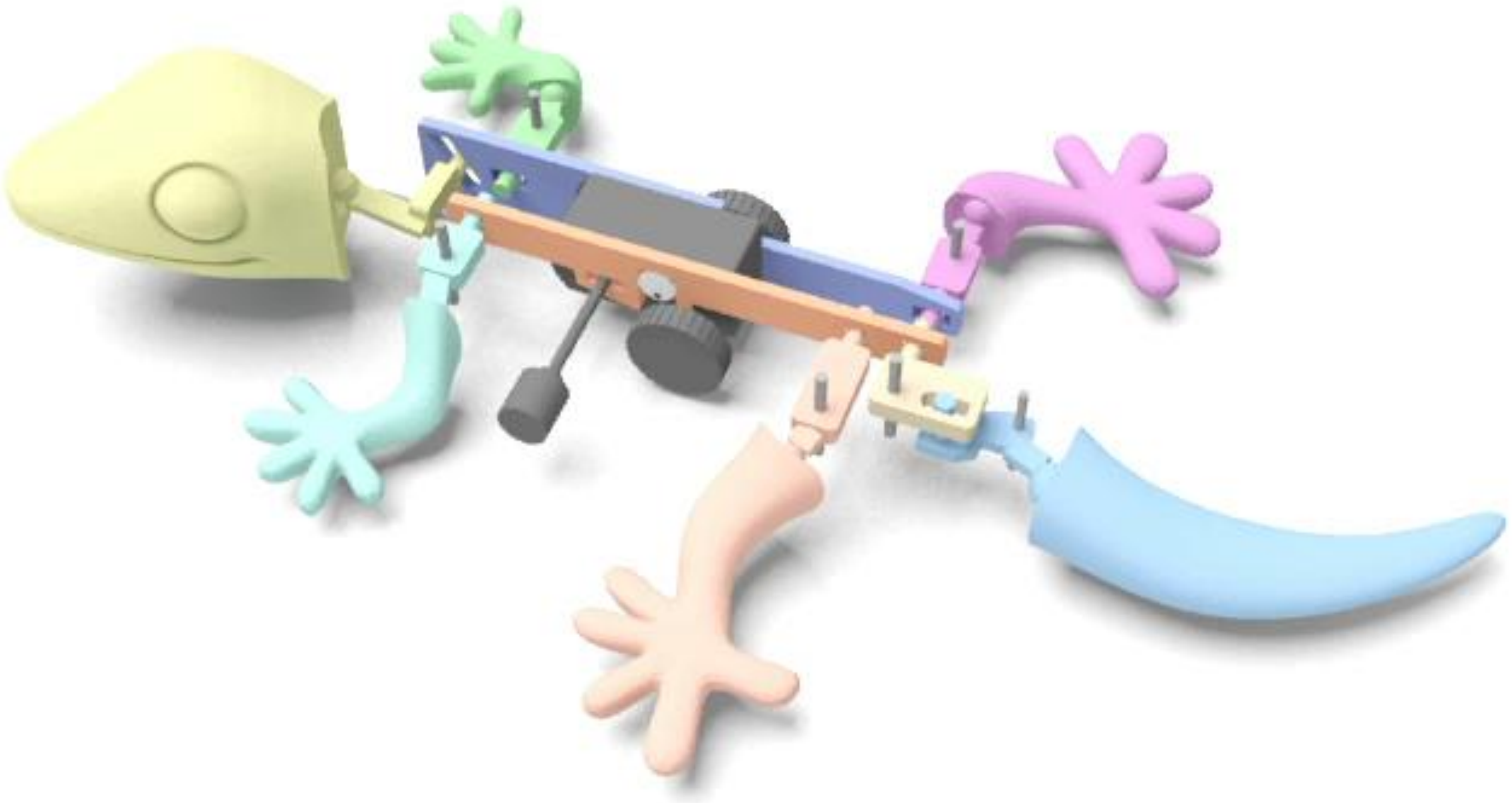


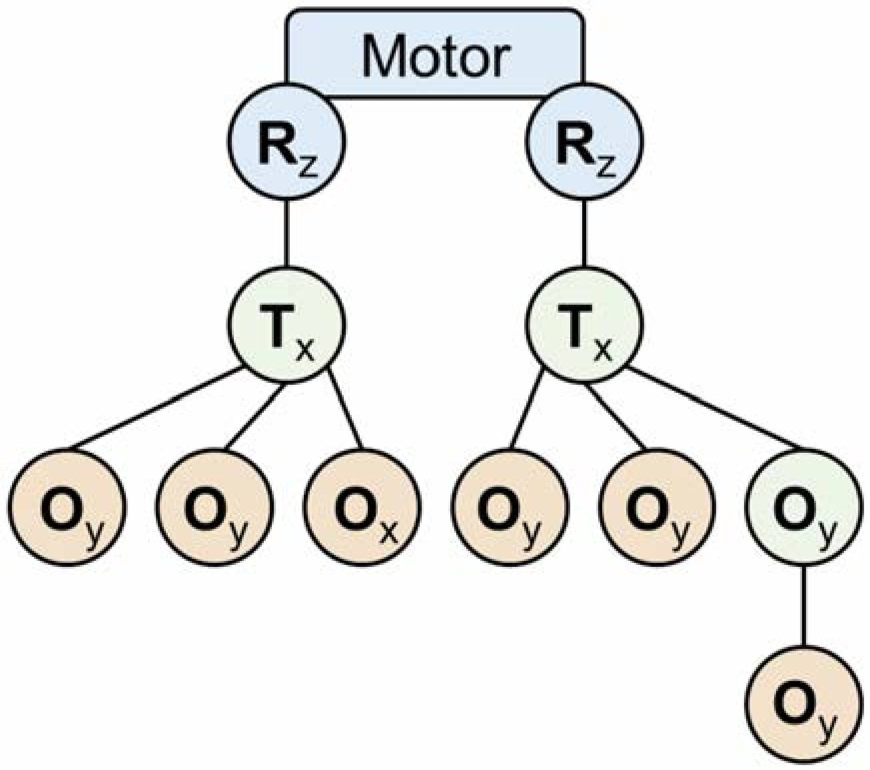


# **Results:** Gecko



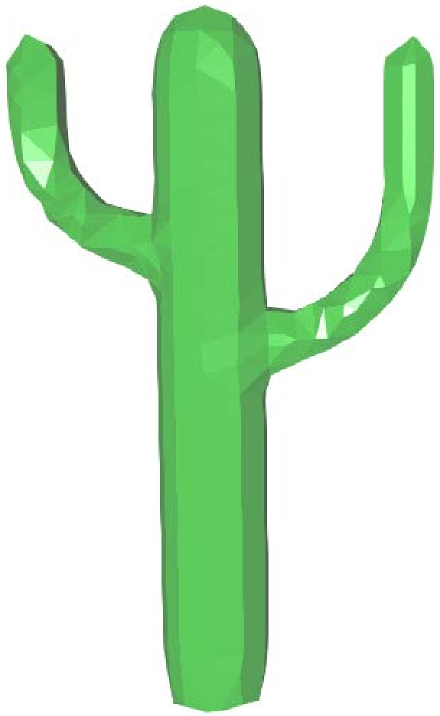
# **Results:** Gecko

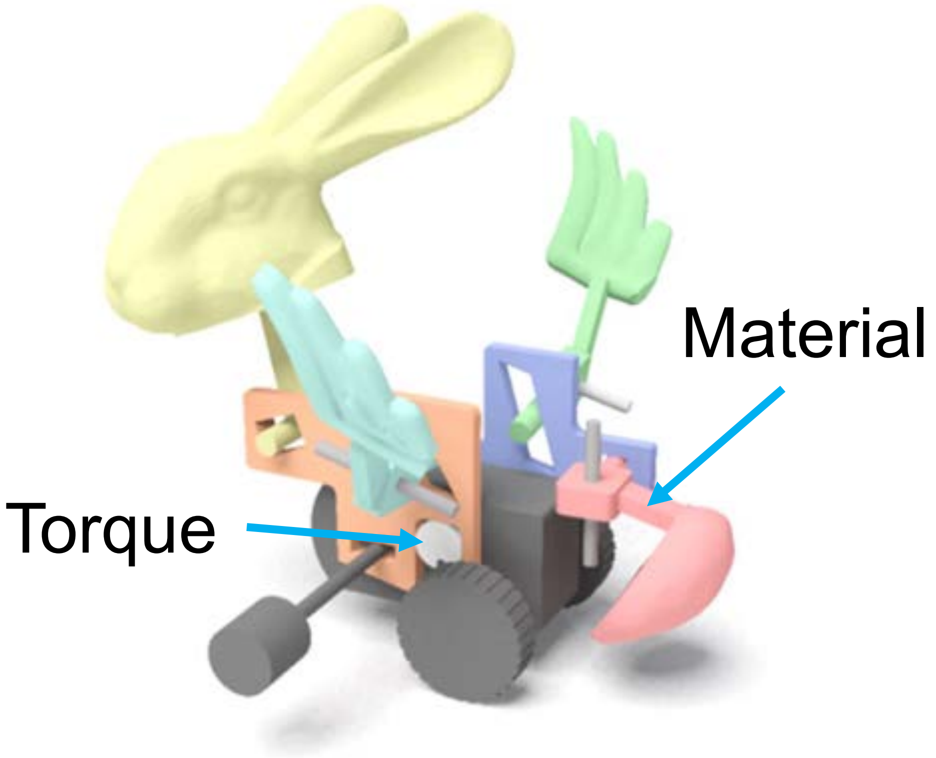




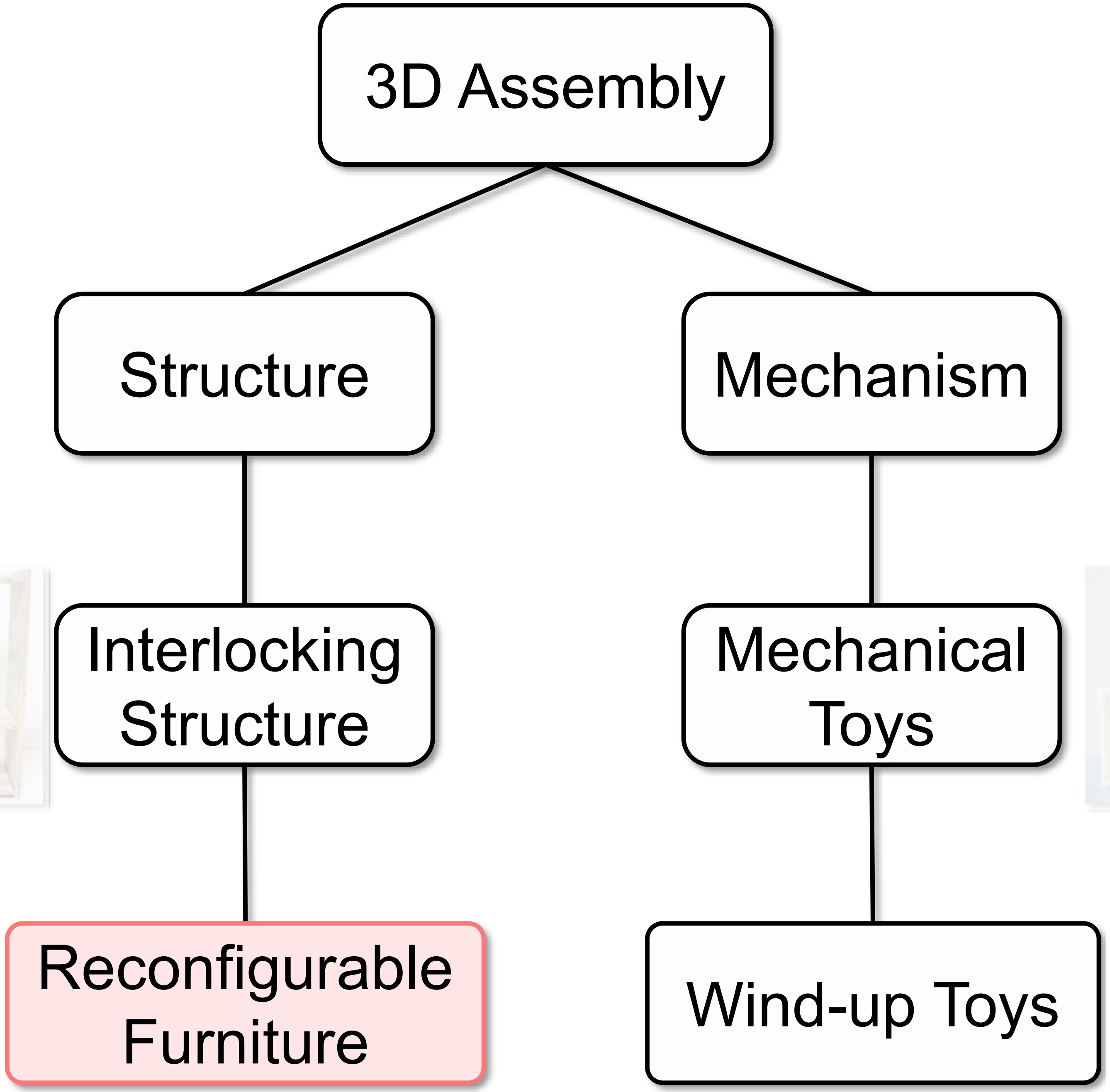
# Limitations and Future Work

- Cannot guarantee a feasible mechanism for arbitrary user inputs
- Cannot guarantee the torque/force of an existing wind-up motor is sufficient to drive our designed mechanism
- Support other kinds of locomotion besides wheeling






# Content


3D Assembly
Structure
Mechanism
Interlocking Structure
Mechanical Toys
Reconfigurable Furniture
Wind-up Toys



# Reconfigurable

Structure that can be transformed into different forms, for use in different situations, e.g., by means of hinges

# Reconfigurable

We may construct **a common set of parts**, and **assemble** the parts into objects of different forms.

a common set of parts


LEGO Creator

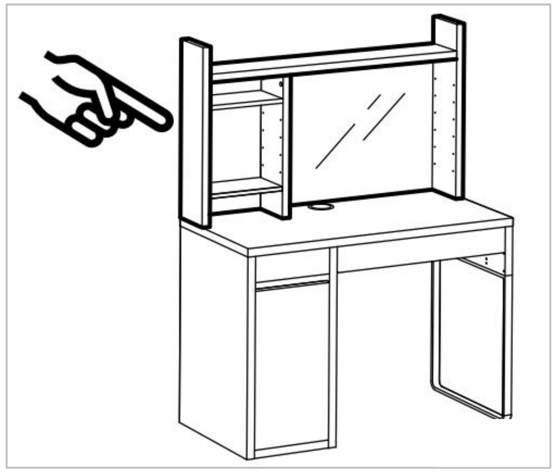

5 In 1 Soft Plastic Nuts Assembled Toy

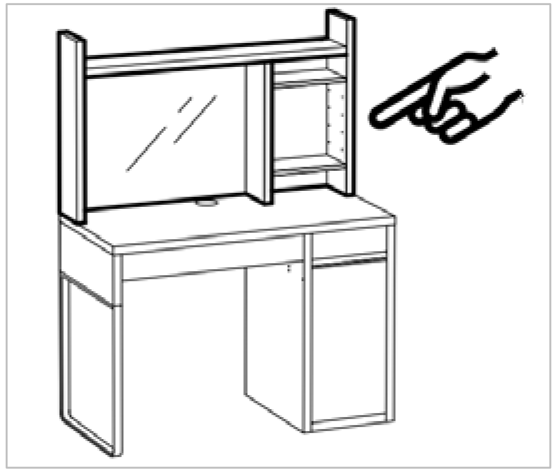
# Reconfigurable Furniture

Furniture assemblies that are reconfigurable and modular…

By T. Maitz of perludi.com

BESTÅ from IKEA

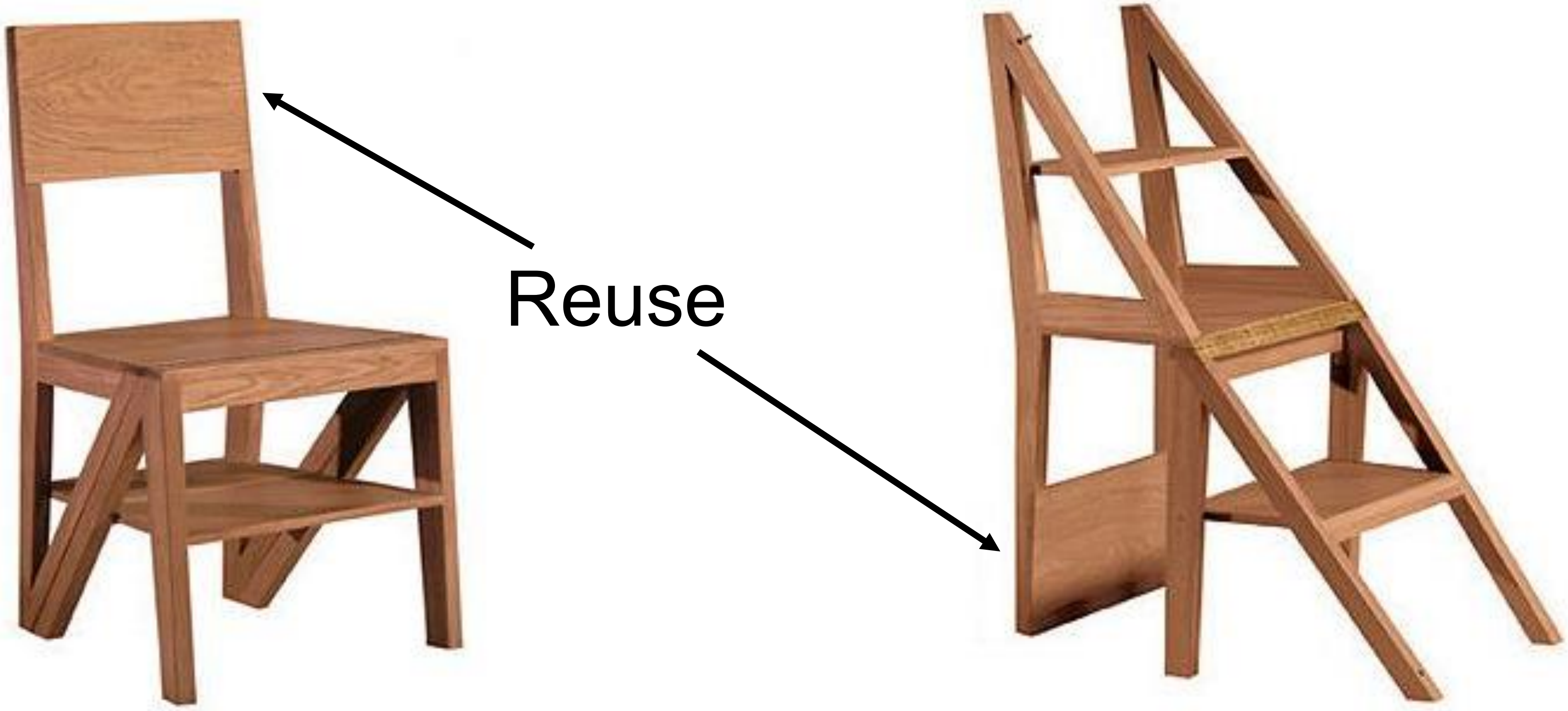




Micke from IKEA

# Advantage #1  Cost-effective

Promote *reuse* of parts and increase their parts utilization



# Advantage #2  Save Space

Save occupied space on usage, storage, and transportation

Non-
Reconfig.


Reconfig.

# Advantage #3 Extensible

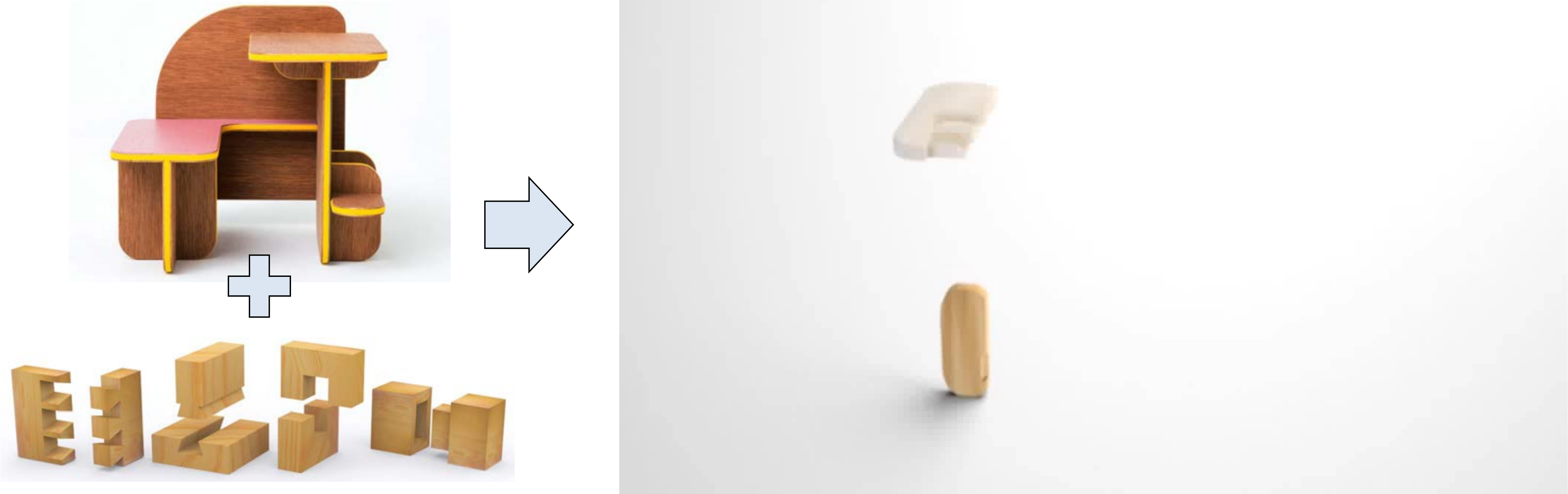
*Extensibility* of the structure by reusing more part instances

Our Result

# In SIGGRAPH 2015 (our previous work)

"Computational interlocking furniture"

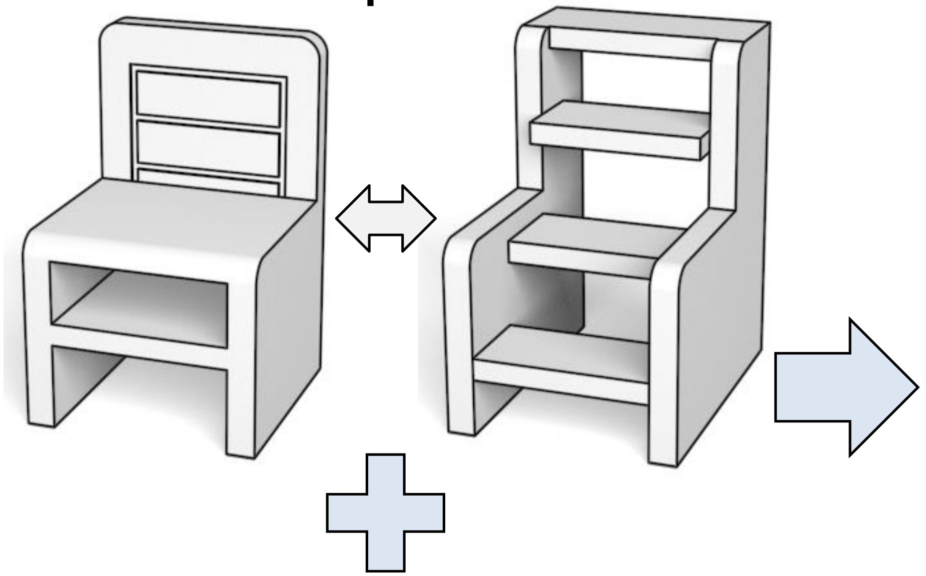
input

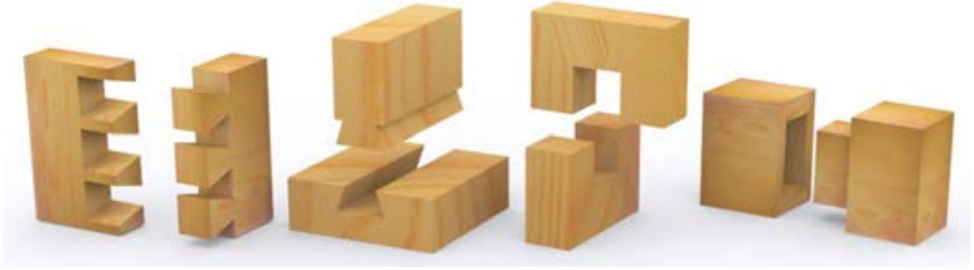


# In SIGGRAPH Asia 2017 (our new work)

“Reconfigurable interlocking furniture”

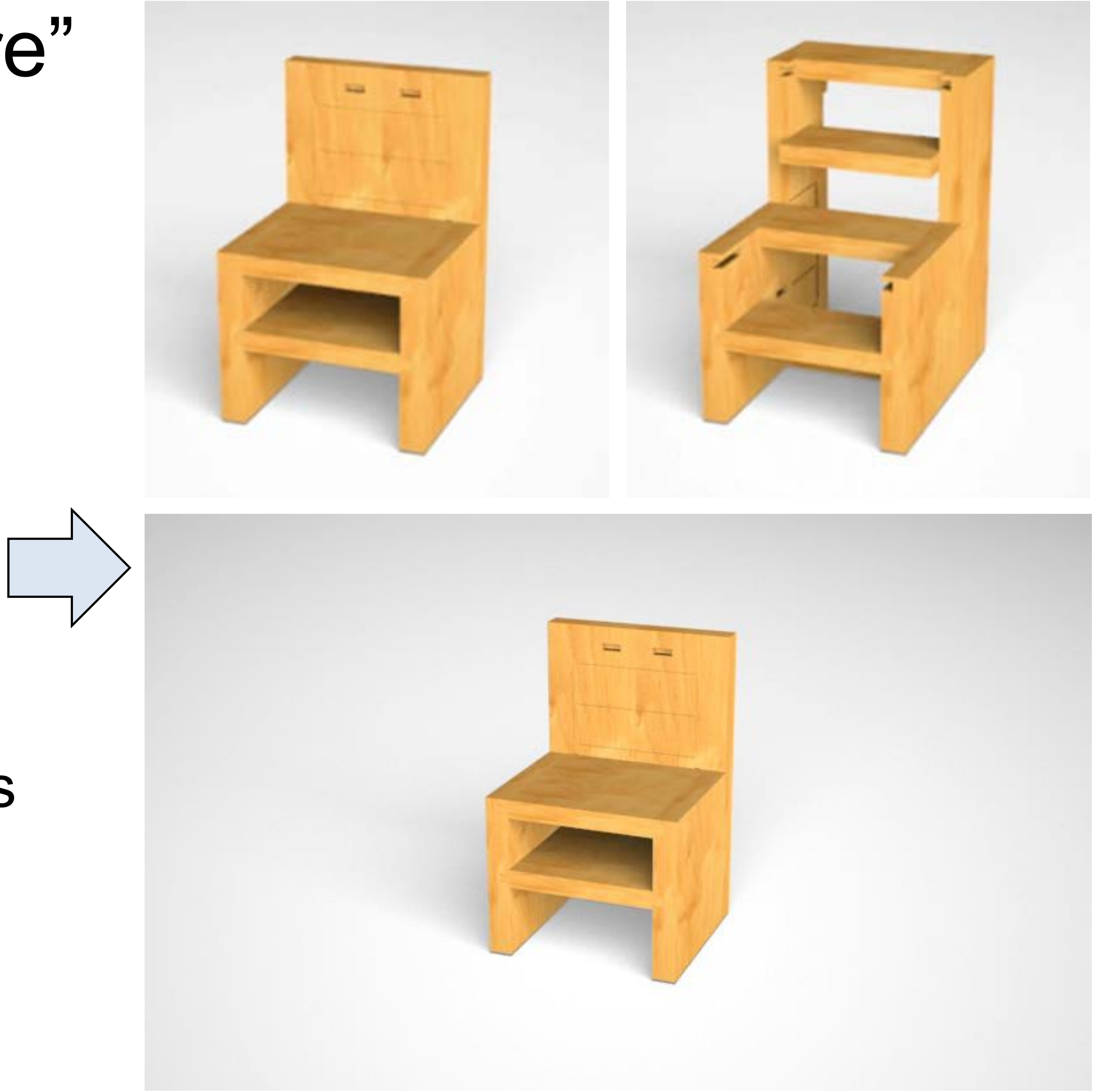
input





a common set of parts

(with joints)



# Related Research Areas

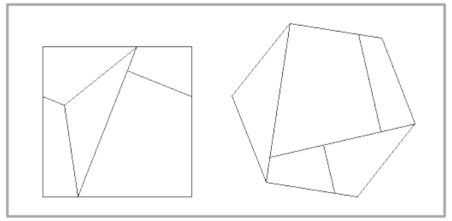

[Zhou et al. 2014]

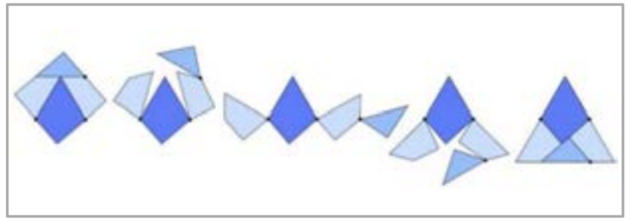

[Sun et al.  2015]


[Li and Hu et al. 2015]


[Garg et al. 2016]

Transformable & Reconfigurable


[Baker 1961]


[Abbott et al. 2012]


[Zhou et al. 2012]


[Duncan et al. 2017]

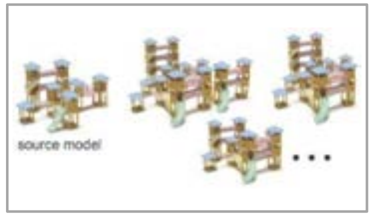
2D/3D Dissection


[Kreavoy et al. 2007]


[Jain et al. 2012]


[Duncan et al. 2016]

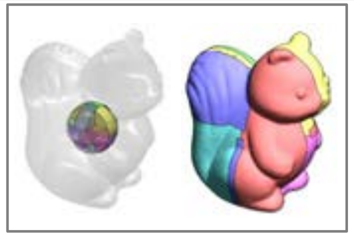

[Zheng et al. 2013]



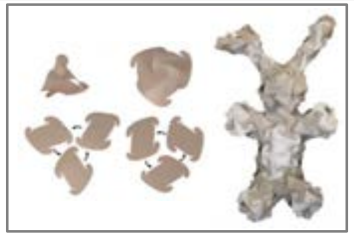
[Liu et al. 2015]


[Kalojanov et al. 2016]

Reusable parts & Reusable substructures


[Xin et al. 2011]


[Song et al. 2012]


[Skouras et al. 2015]


[Fu and Song et al. 2015]

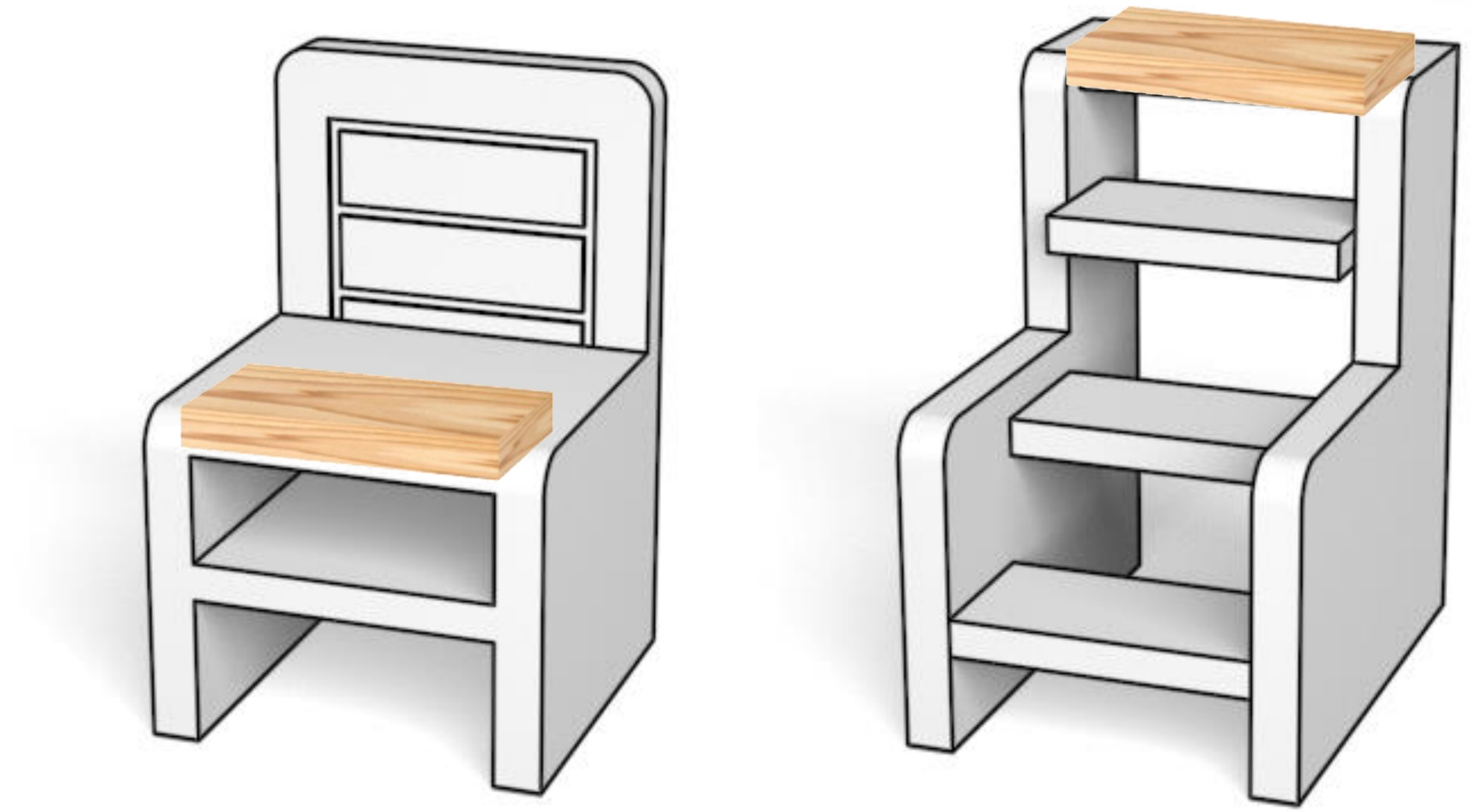

[Song et al. 2016]


[Yao et al. 2017]

Interlocking Assemblies

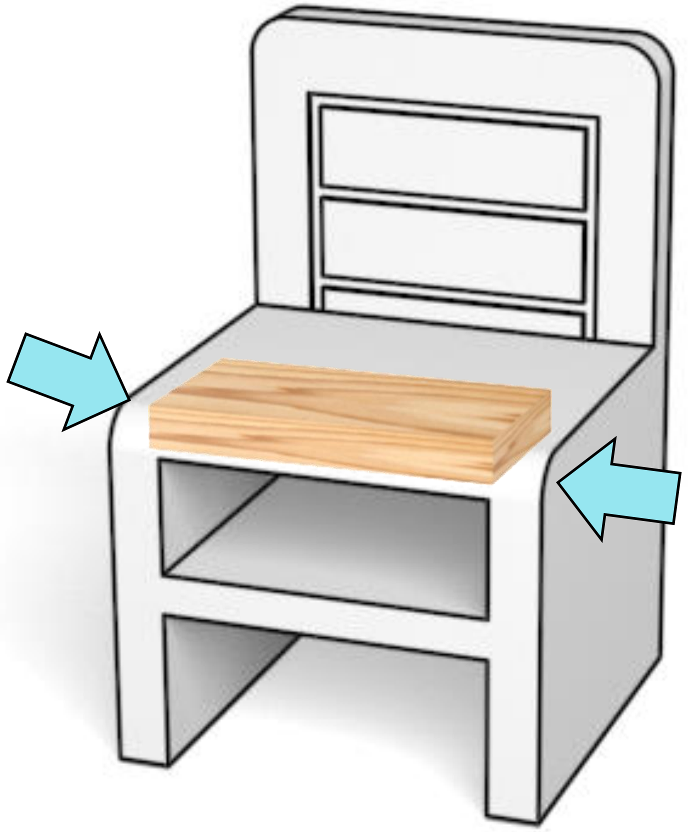
# Challenge #1: Reconfigurable Interlocking Furniture

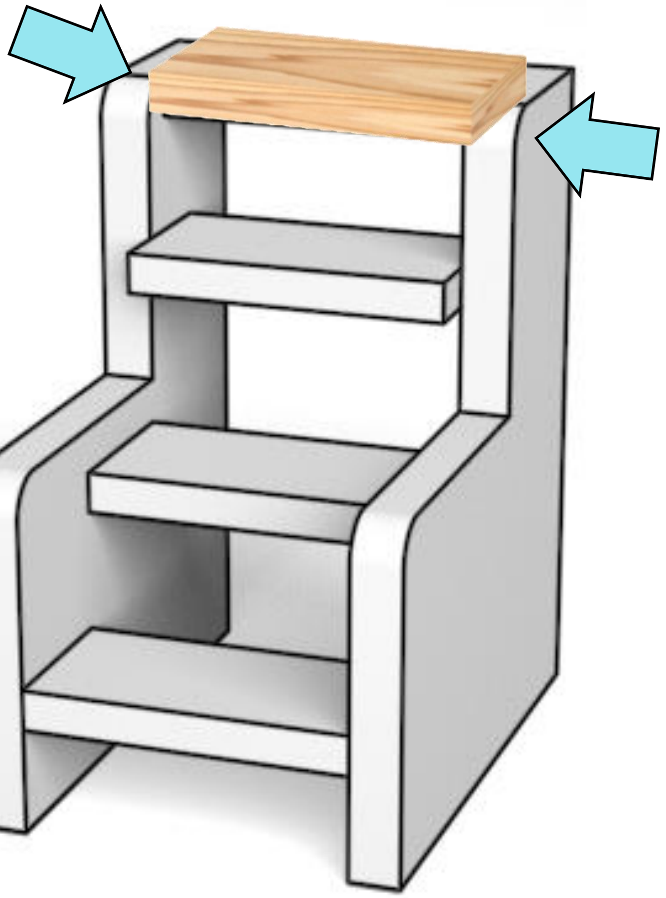
**Compatible Parts:** for a part to be reusable, it has to fit to different designs with exactly the same dimension and shape



# Challenge #2: Reconfigurable Interlocking Furniture

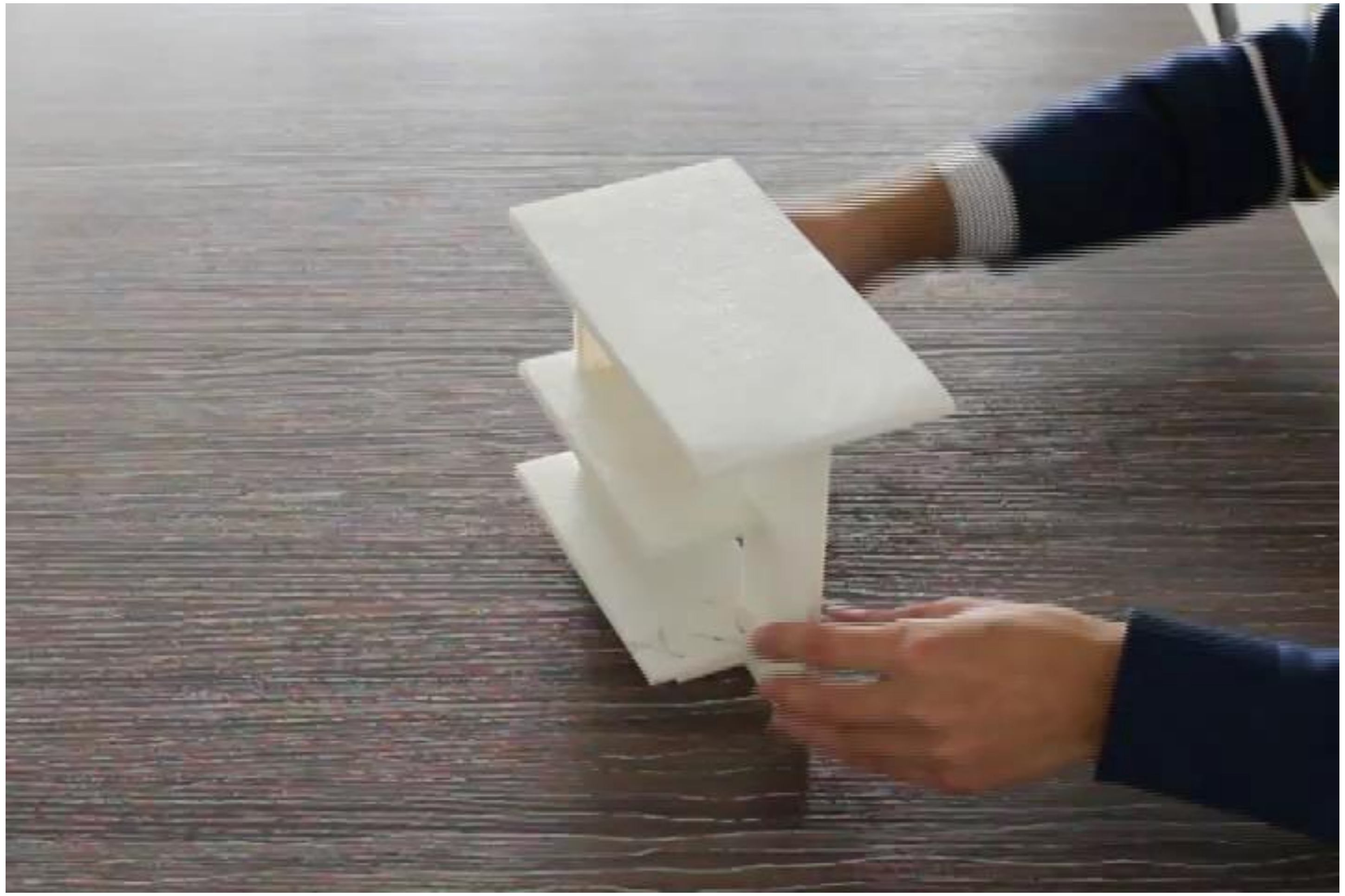
**Compatible Joints:** the joints on the part should allow the part to connect different neighboring parts in different designs





# Challenge #3: Reconfigurable Interlocking Furniture

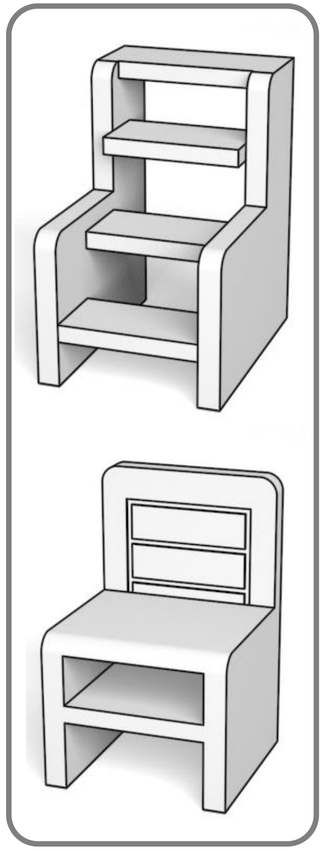
**Steady assembly**: we should assemble the parts with interlocking, and be able to repeatedly disassemble and re-assemble them



Steady assembly based on **mechanical interlocking**

# Overview of Our Approach

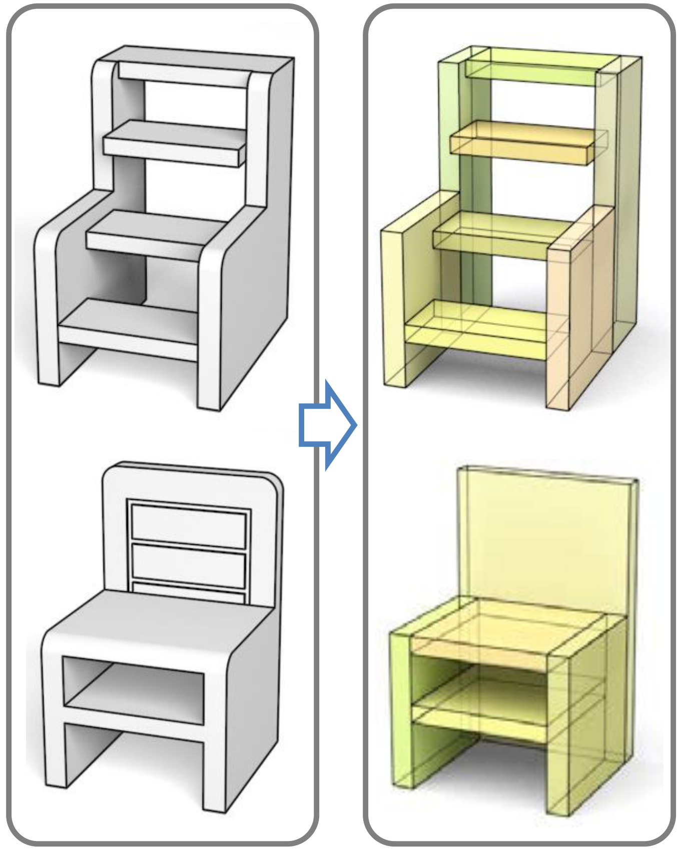
Starting from the input designs…



input
designs

# Overview of Our Approach

Approximate inputs by cages and identify geometric semantics

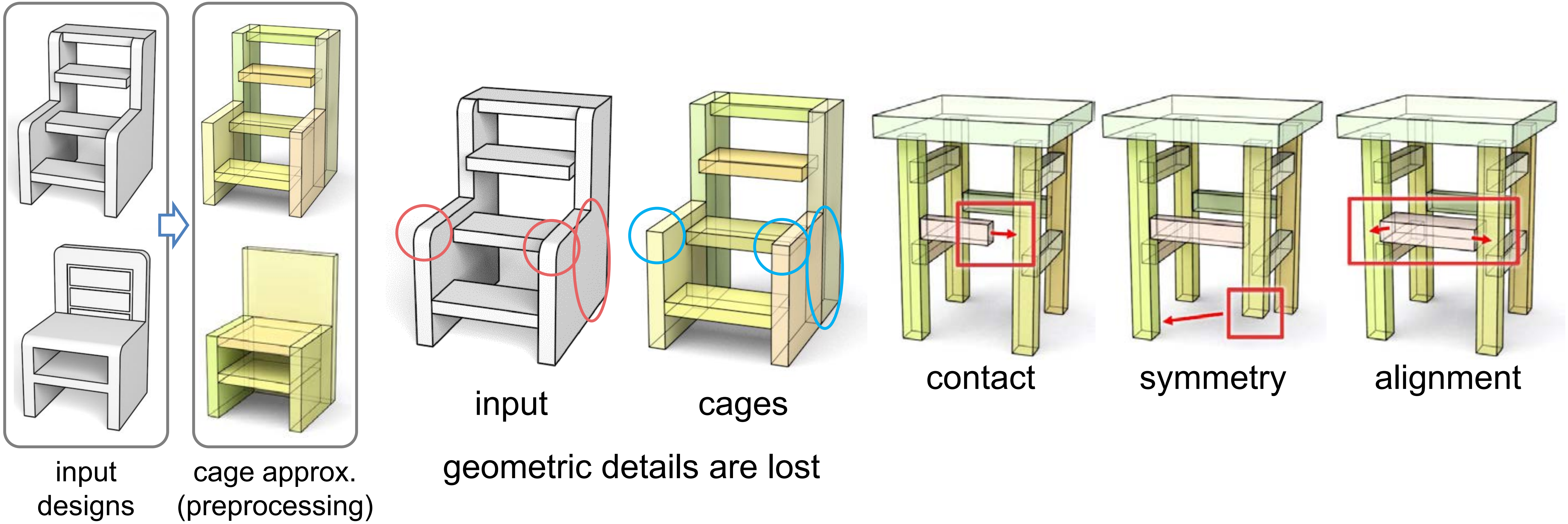


input designs

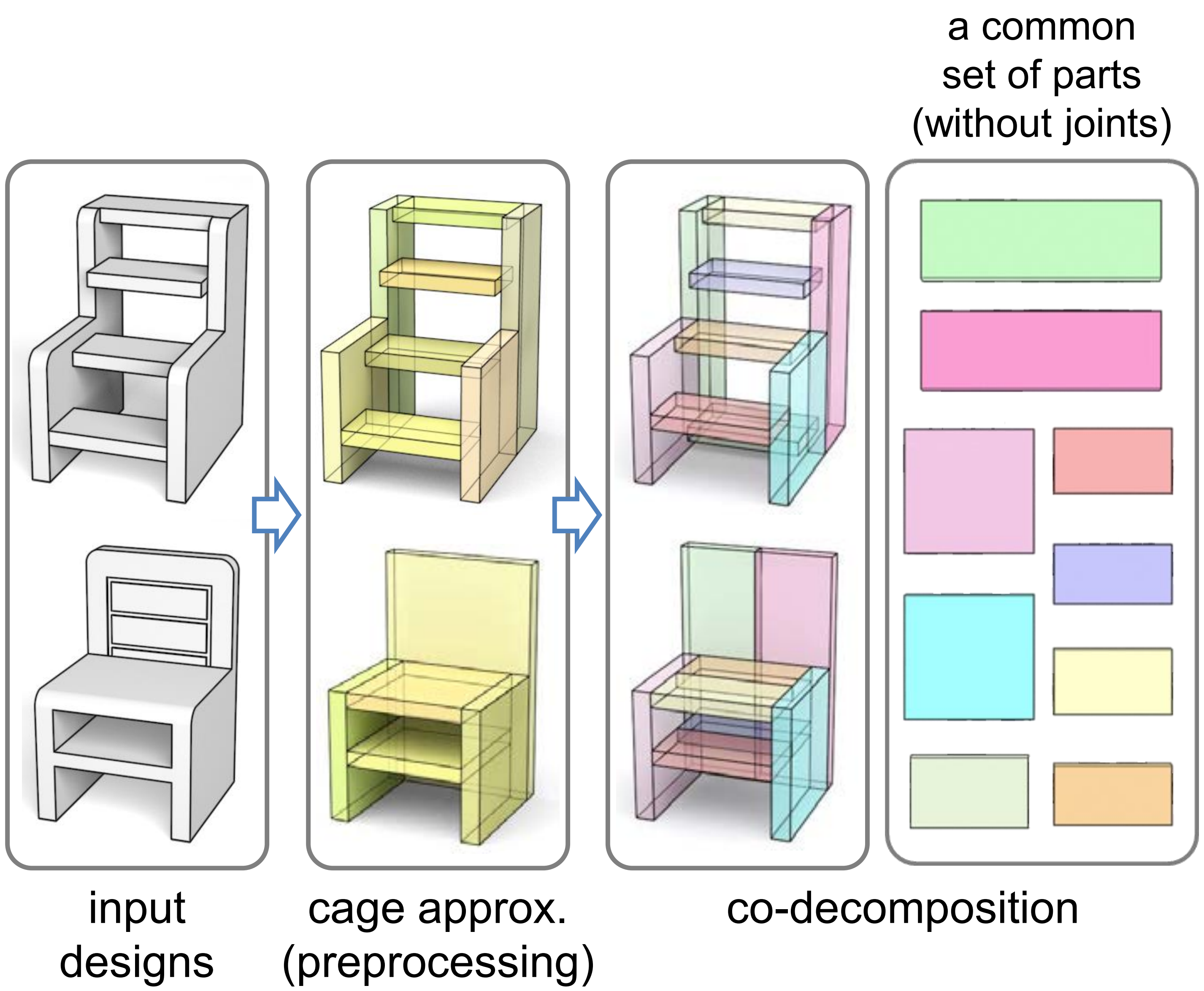
cage approx. (preprocessing)

# Overview of Our Approach

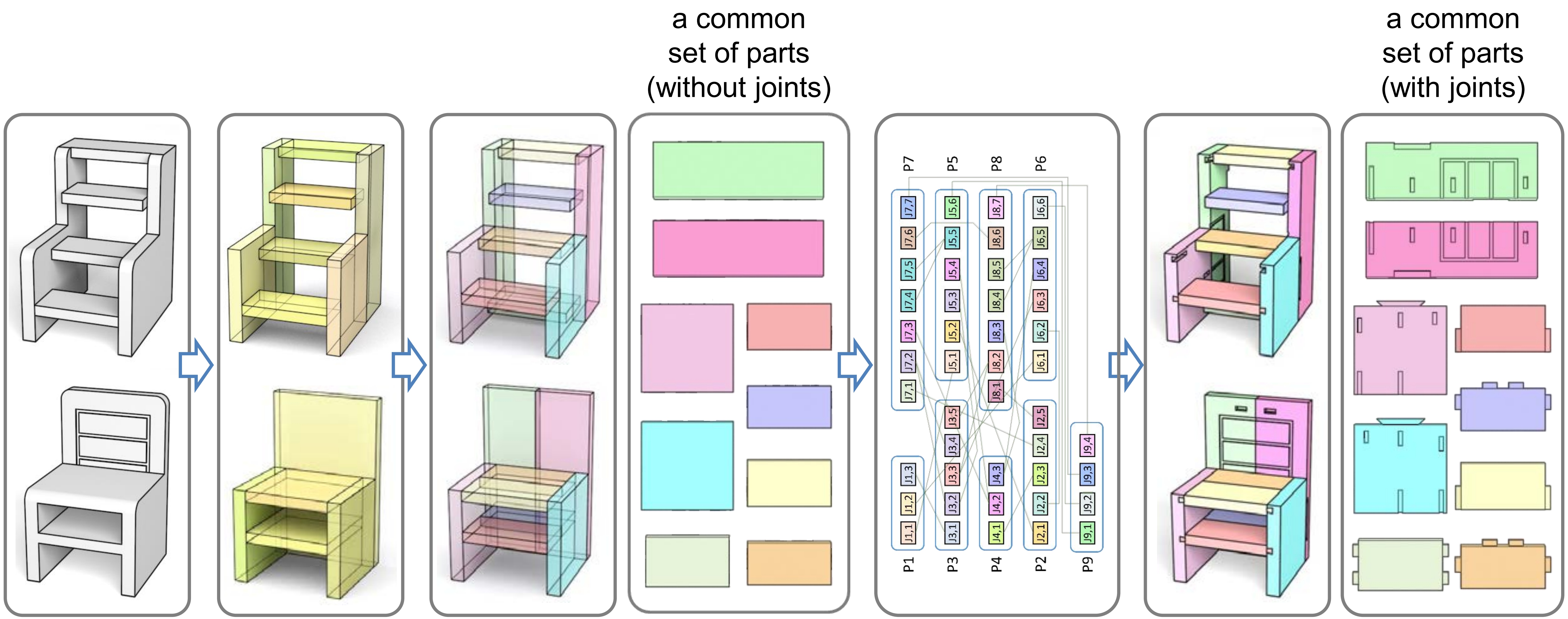
Approximate inputs by cages and identify geometric semantics



# Overview of Our Approach



# Overview of Our Approach




input designs

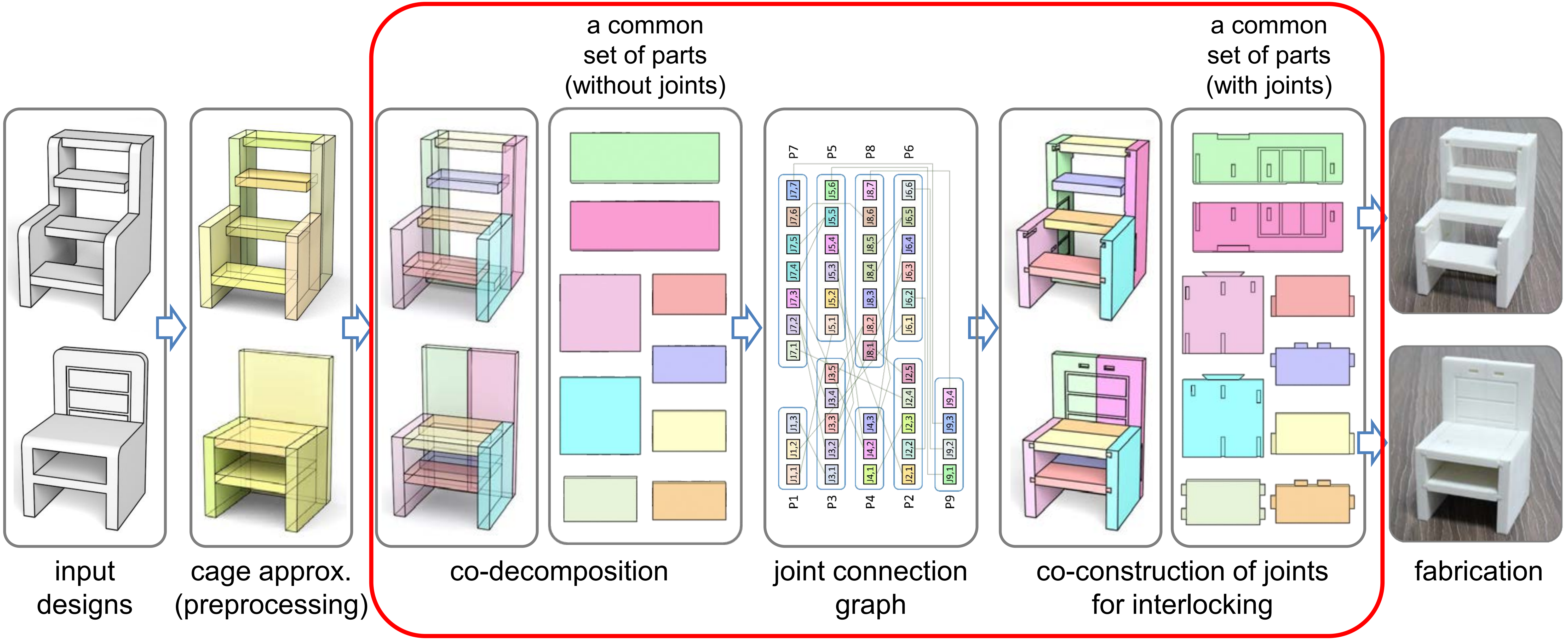
cage approx. (preprocessing)

co-decomposition

joint connection graph

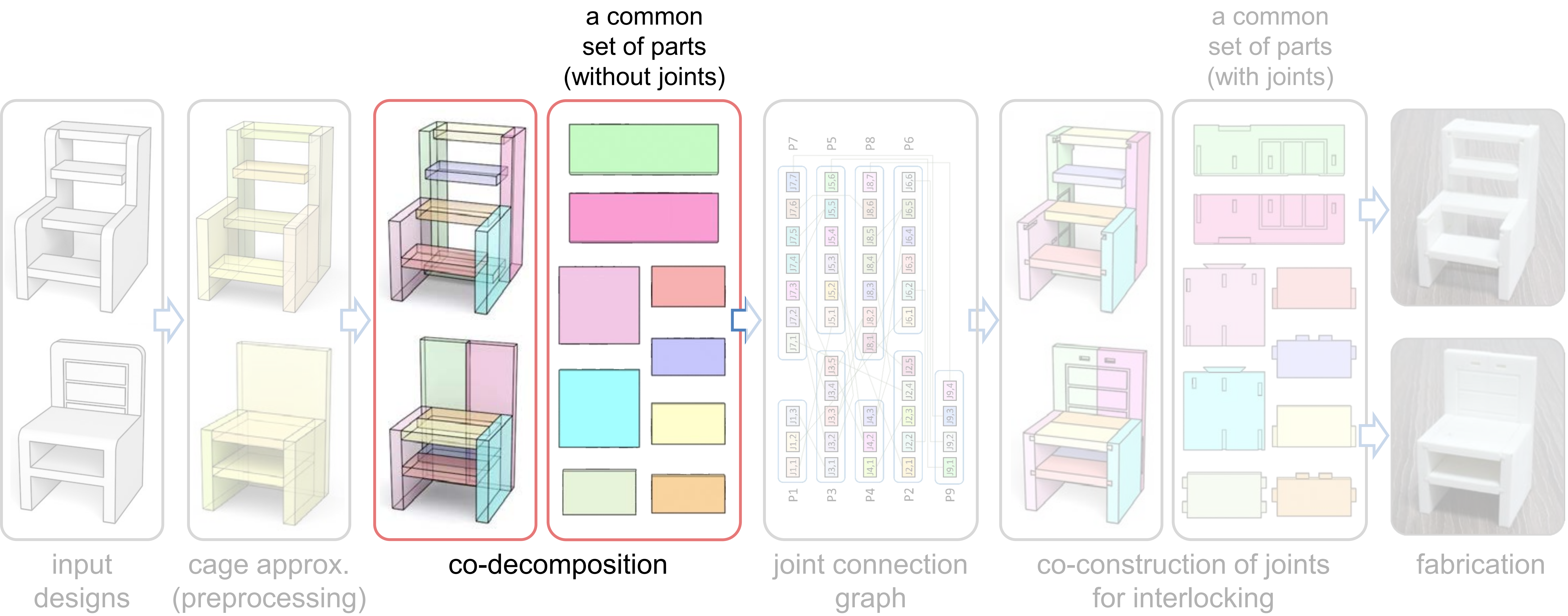
co-construction of joints for interlocking

# Overview of Our Approach



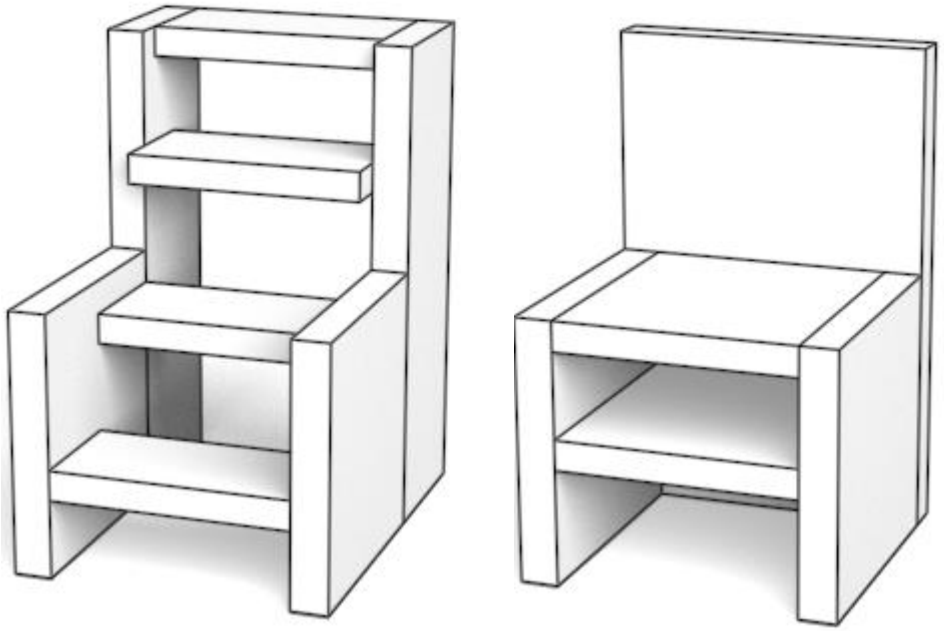

our key contributions

# Overview of Our Approach



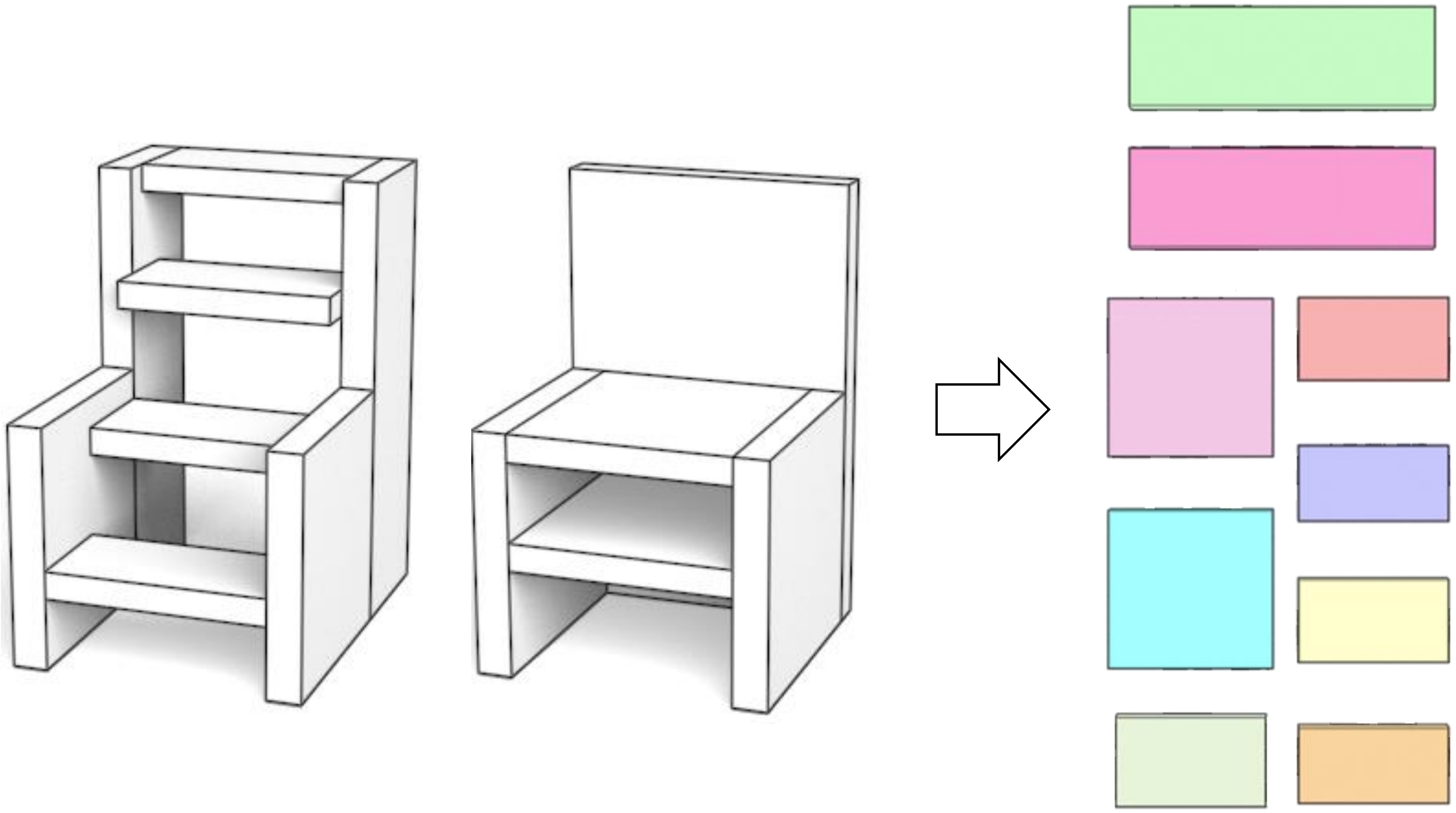
# **Co-decomposition:** Goal

Given input designs approximated with cages,



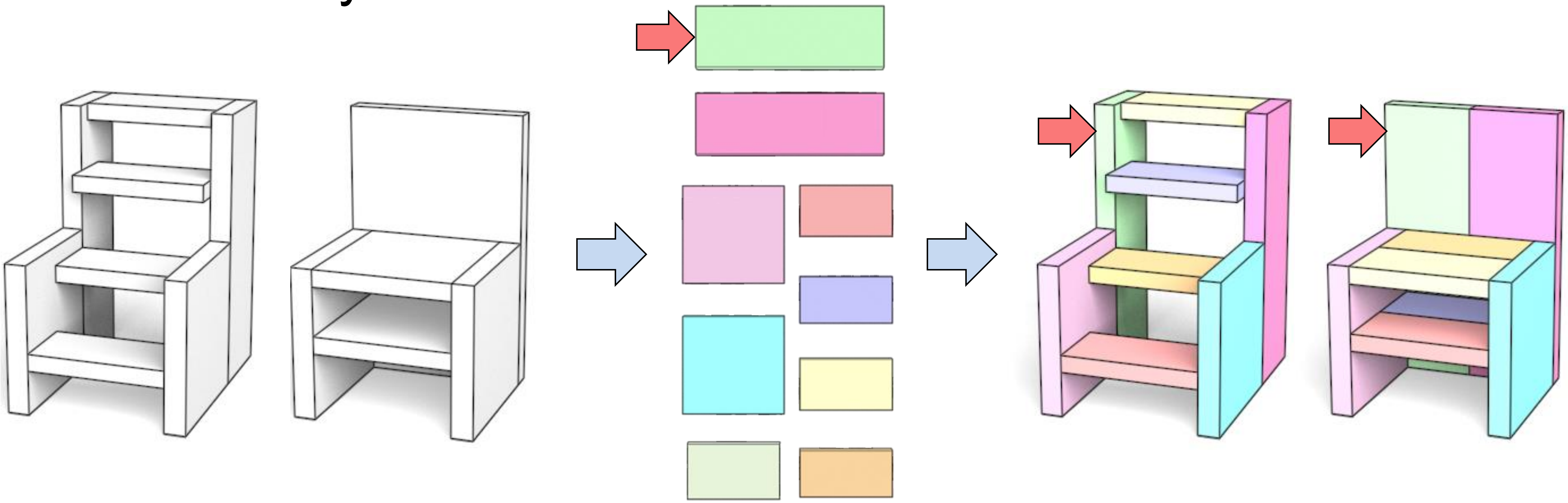
# **Co-decomposition:** Goal

Given input designs approximated with cages, our goal is to construct <u>a common set of parts</u>



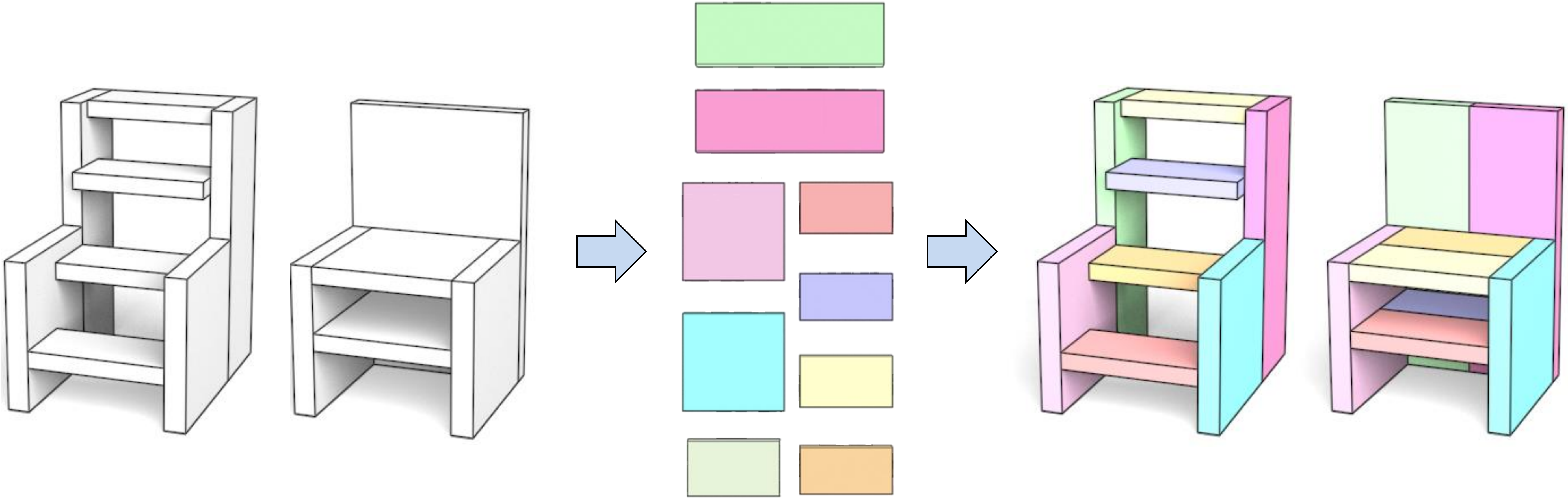
# **Co-decomposition:** Goal

Given input designs approximated with cages, our goal is to construct <u>a common set of parts</u> that can be reused to build the assembly structures.



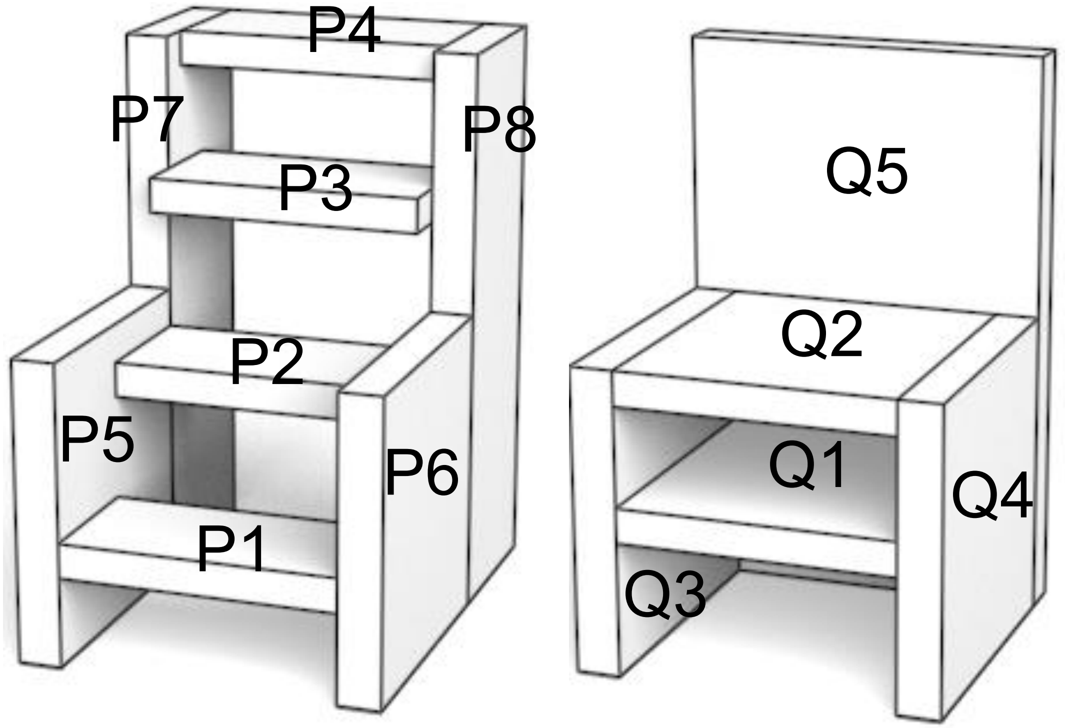
# **Co-decomposition:** Requirements

- **Parts utilization**: maximize the reuse of parts across designs
- **Easy assembly**: minimize the total number of common parts
- **Design fidelity**: minimize local modifications on input designs

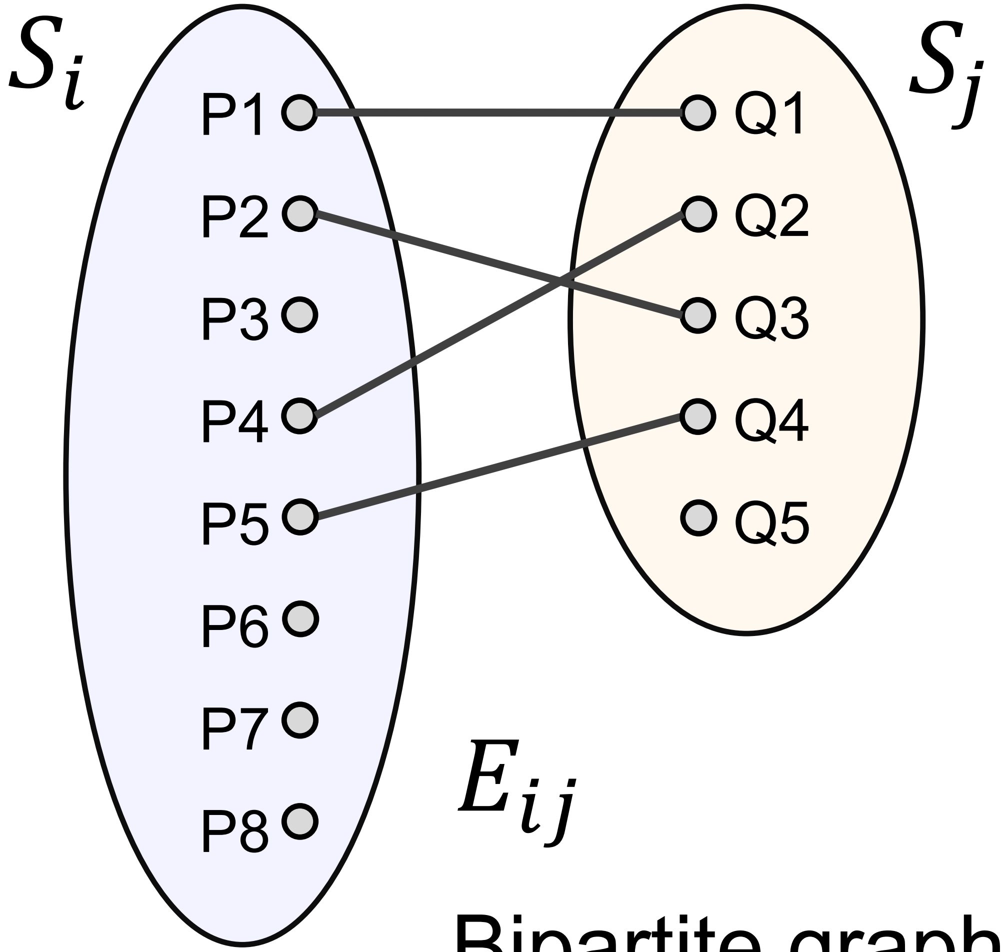


# **Co-decomposition:** Formulation

We formulate co-decomposition as an edge construction problem on dynamic bipartite graphs:



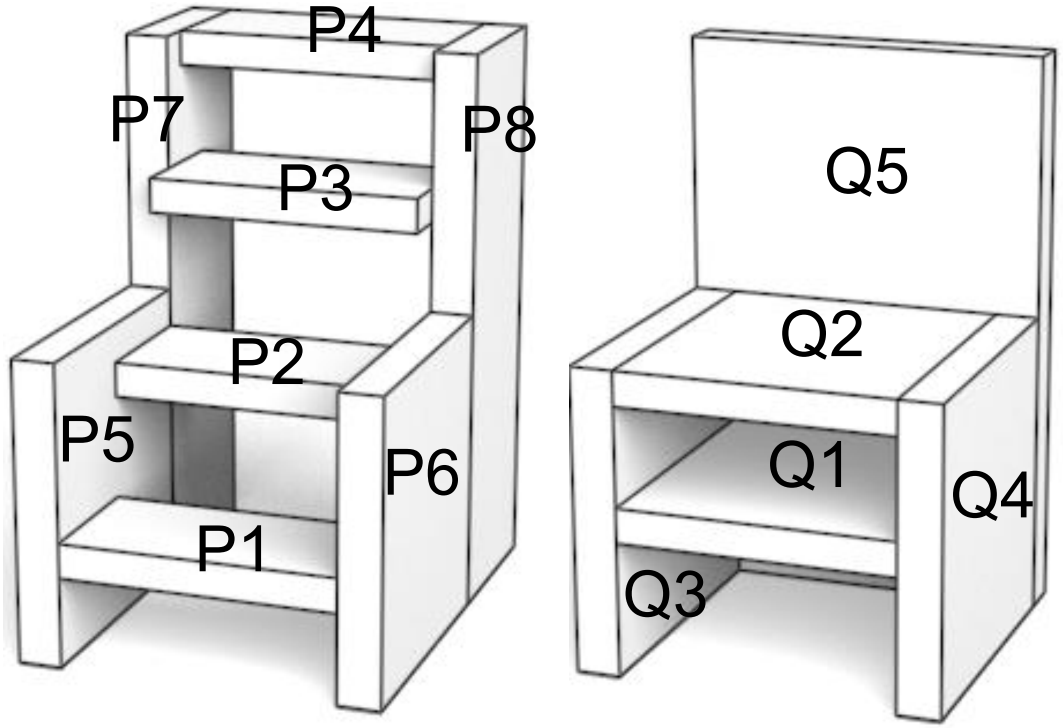

Designs



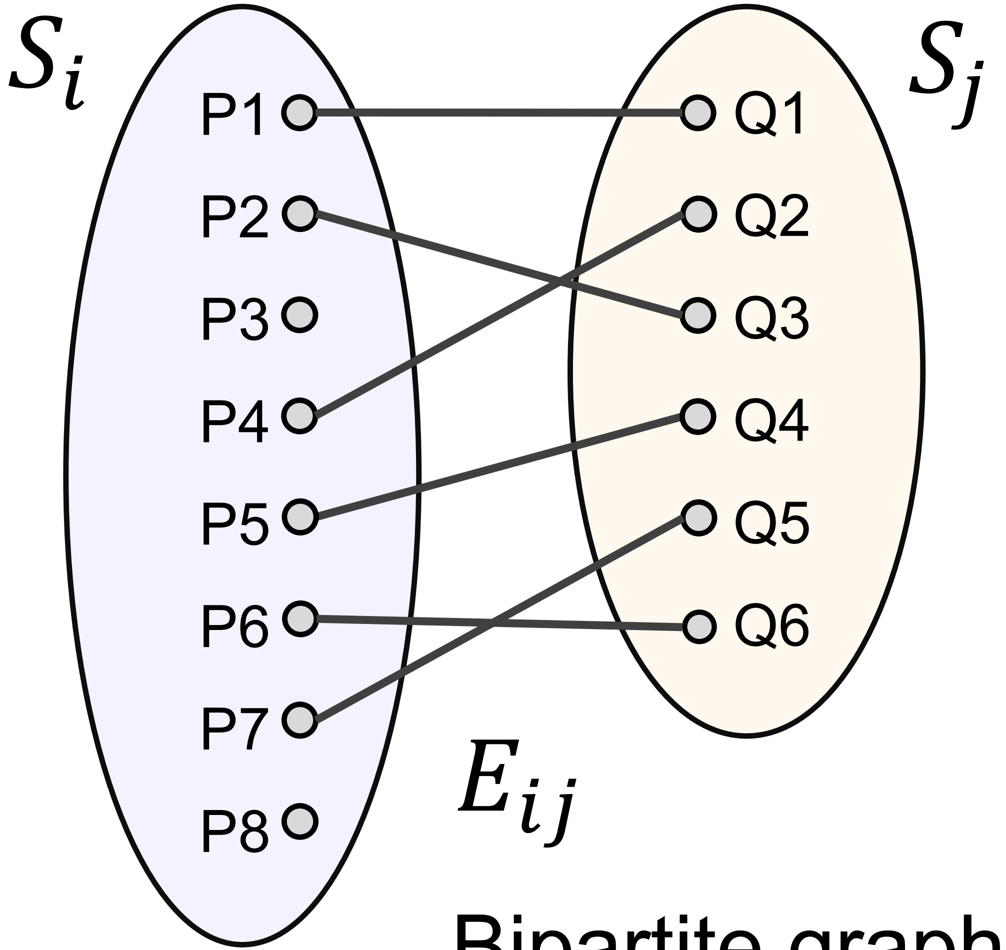

Bipartite graphs

# **Co-decomposition:** Formulation

We formulate co-decomposition as an edge construction problem on dynamic bipartite graphs:




Designs




Bipartite graphs

# **Co-decomposition:** Formulation

Objective function for the edge construction problem:

$$\max \sum_{i,j\ s.t.\ i\neq j} \left[ \omega_1 C_1(\mathbf{S}_i, \mathbf{S}_j) - \omega_2 C_2(\mathbf{S}_i, \mathbf{S}_j) \right] + \sum_i \omega_3 C_3(\mathbf{S}_i),$$

$$\text{with} \quad C_1(\mathbf{S}_i, \mathbf{S}_j) = |\mathbf{E}_{ij}|,$$

Maximize parts reuse

$$C_2(\mathbf{S}_i, \mathbf{S}_j) = |\mathbf{S}_i| + |\mathbf{S}_j| - |\mathbf{E}_{ij}|,$$

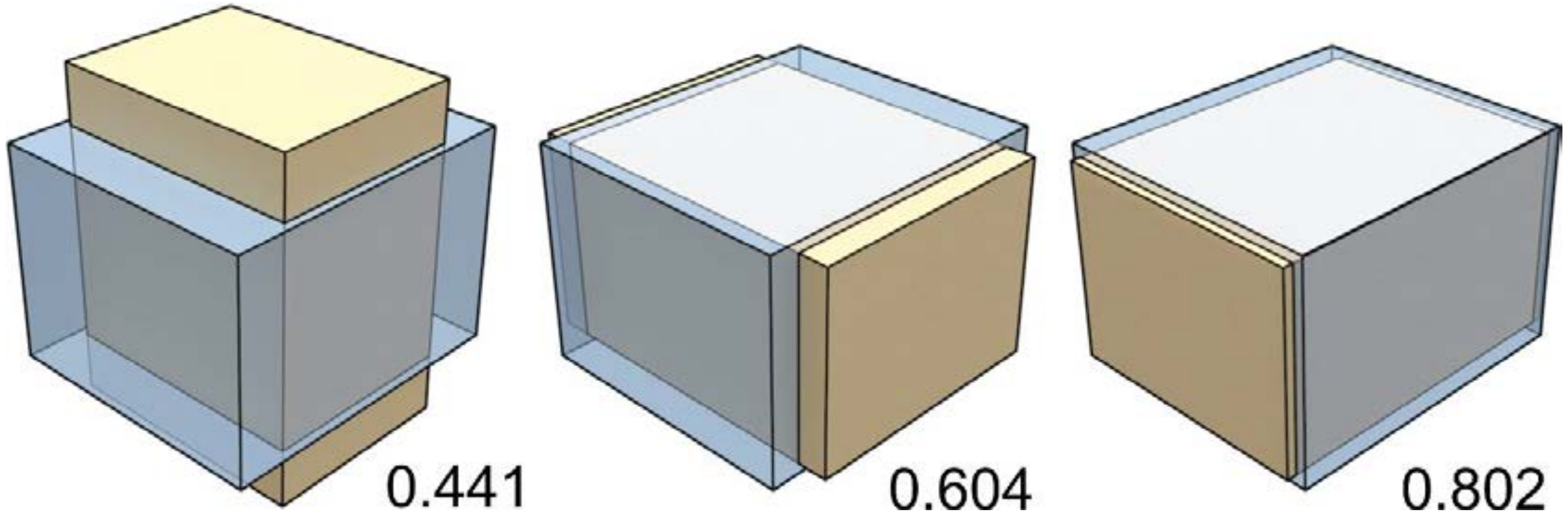
Smaller common set of parts

$$\text{and} \quad C_3(\mathbf{S}_i) = \sum_k \frac{\mathrm{vol}(B_k^i \cap \bar{B}_k^i)}{\mathrm{vol}(B_k^i \cup \bar{B}_k^i)},$$

Small modifications on designs

# **Operation #1:** Measure Cage Similarity

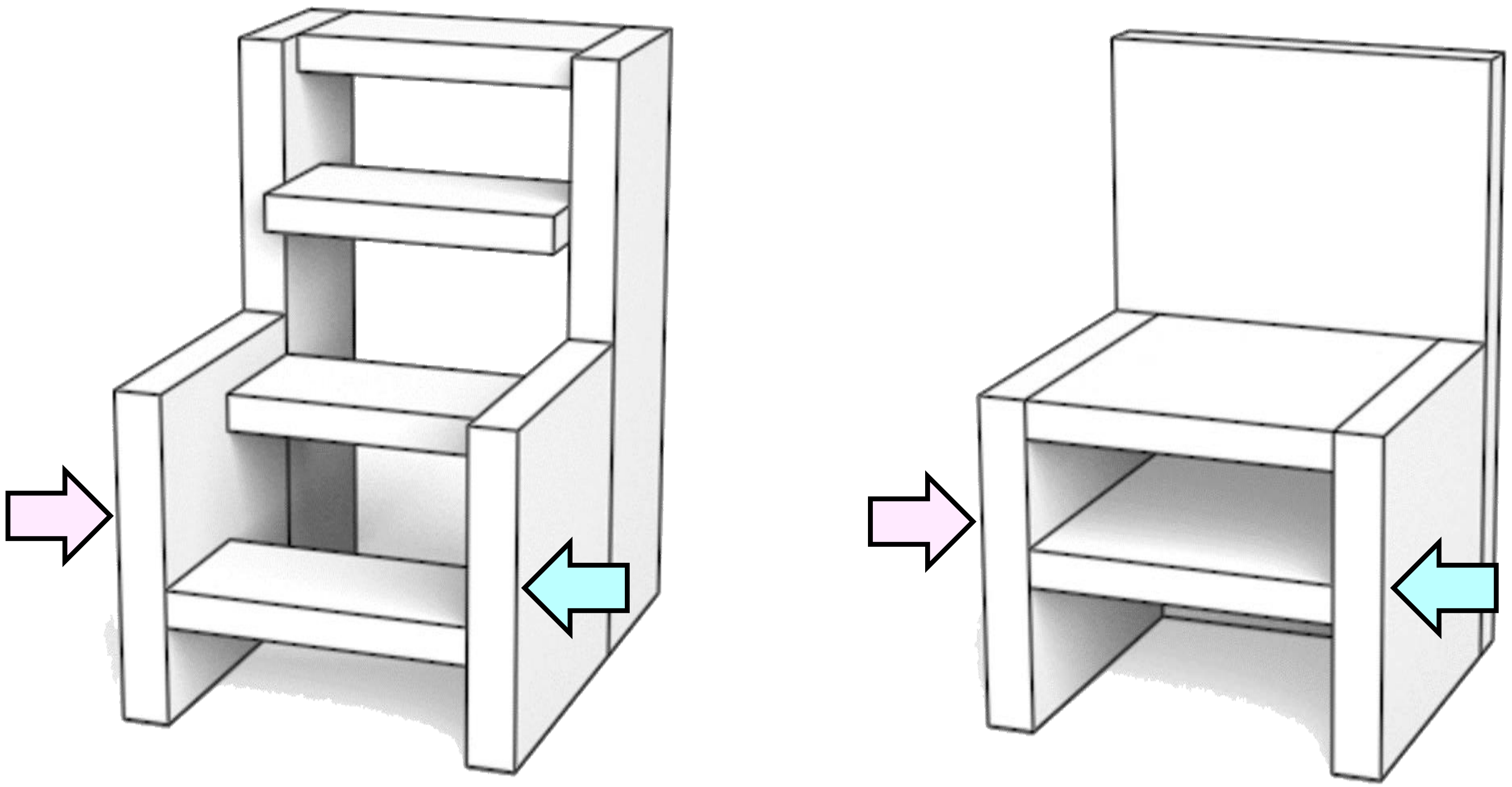
- The goal is to identify similar cages for modification such that small modification can result in a correspondence
- Cage similarity measure: $\mathbb{S}(B_1, B_2) = \max_i \frac{\text{vol}(\ B_1 \cap R_i(B_2)\ )}{\text{vol}(\ B_1 \cup R_i(B_2)\ )}$




$\mathbb{S}(B_1, B_2)$ = 1 indicates a correspondence

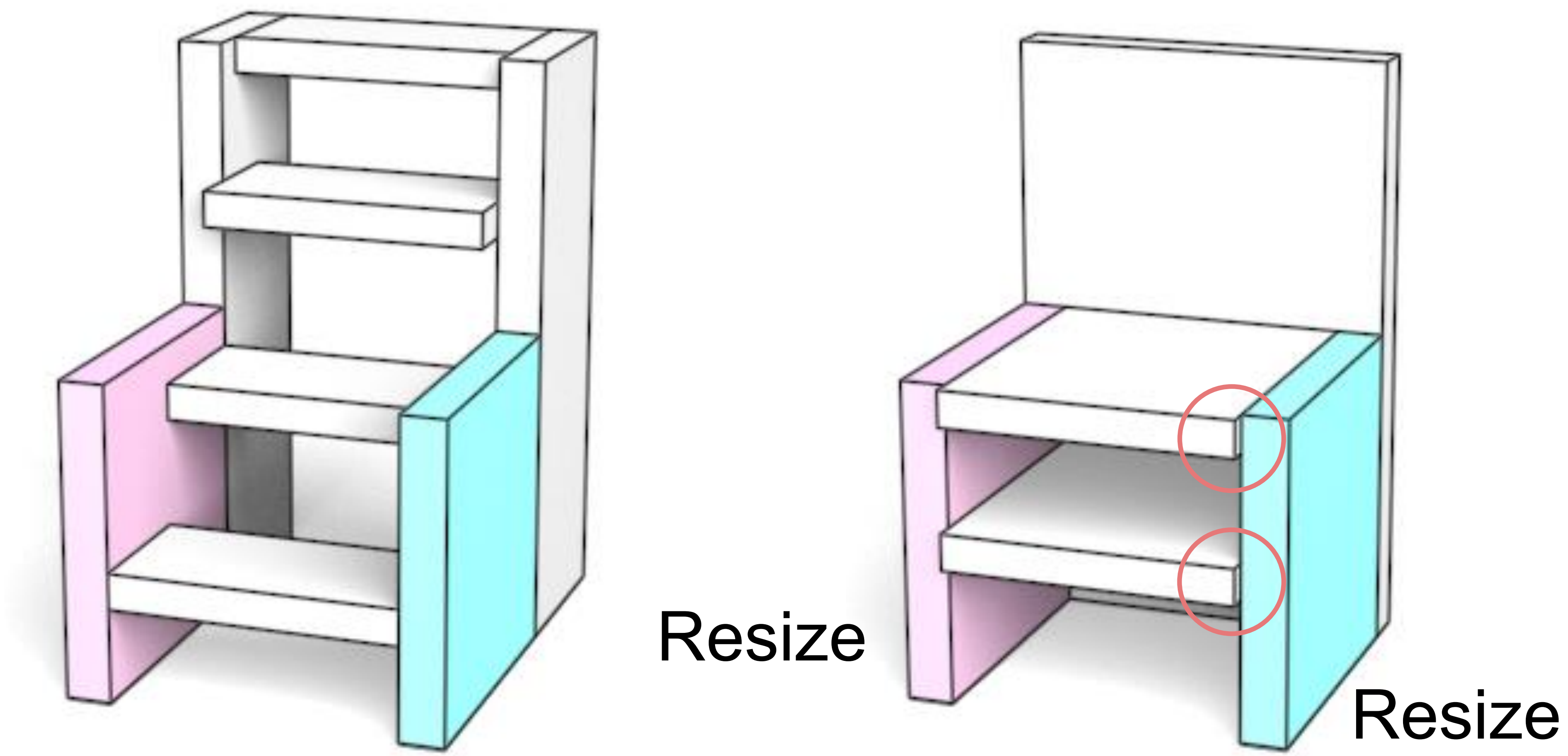
# **Operation #2:** Resize Cages

Form cage correspondences by resizing some of them



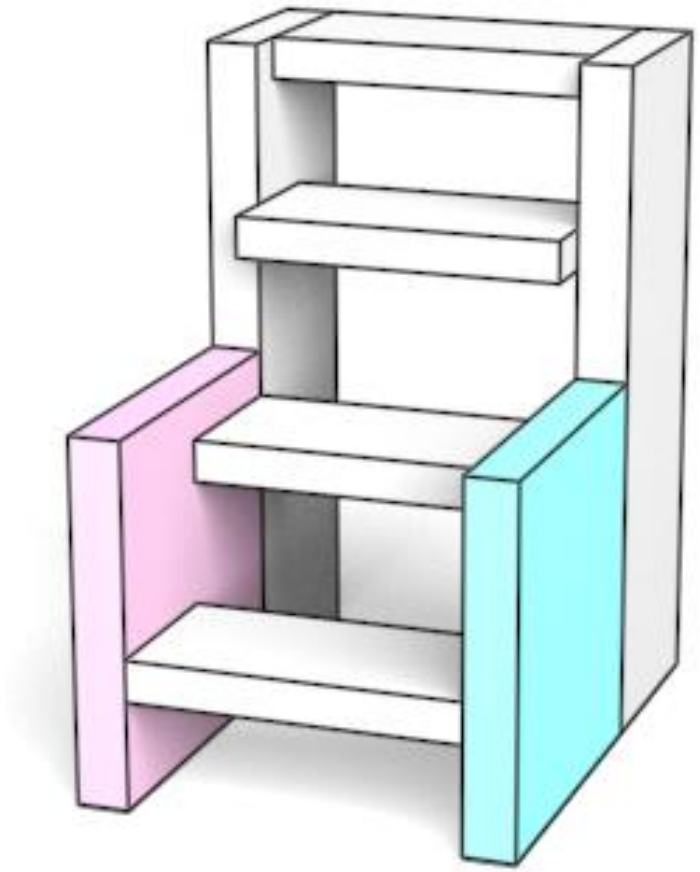
# **Operation #2:** Resize Cages

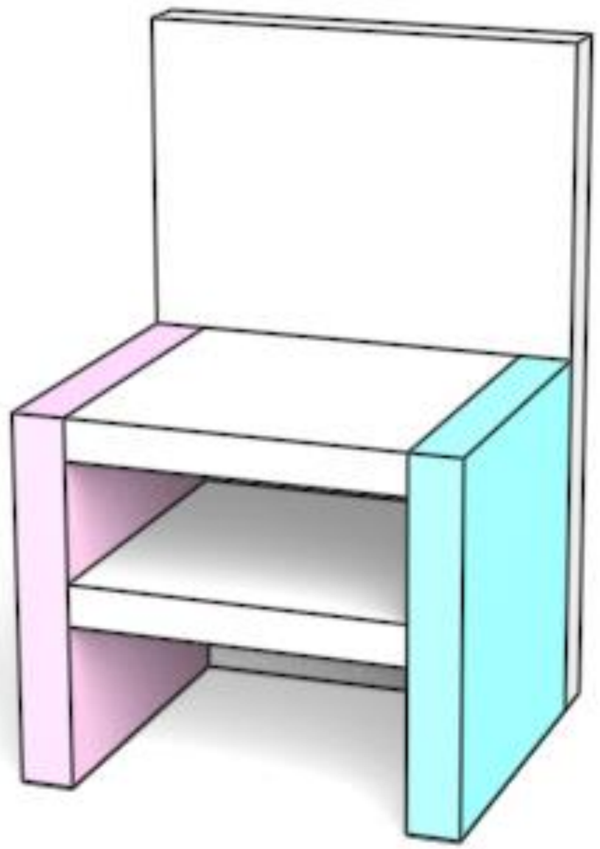
Form cage correspondences by resizing some of them



# **Operation #2:** Resize Cages

Preserve geometric semantics after the cage resizing

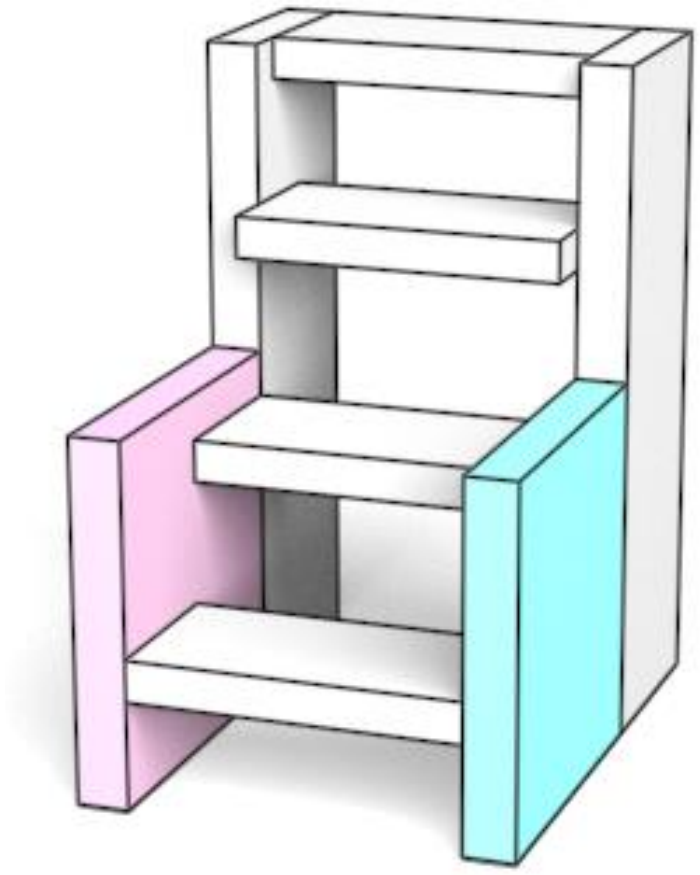


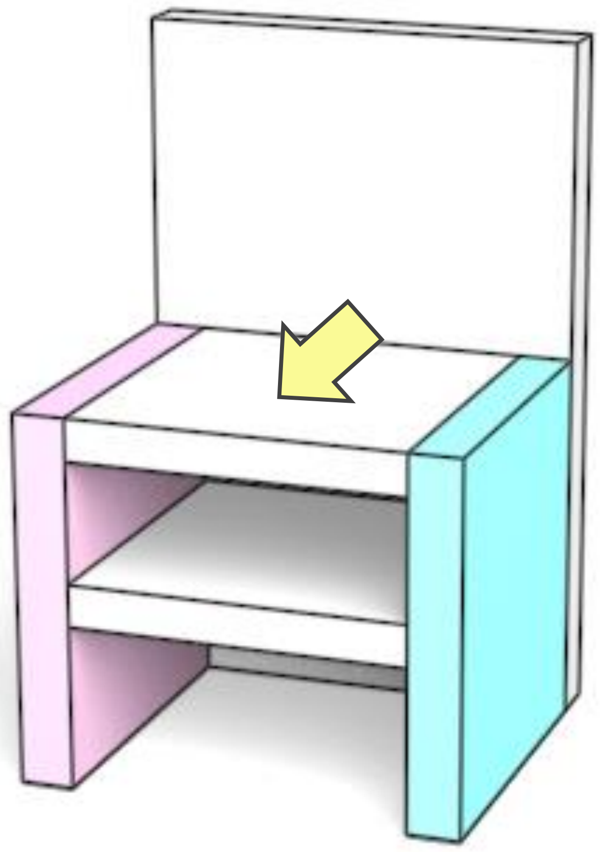


Preserve
**Alignment**

# **Operation #2:** Resize Cages

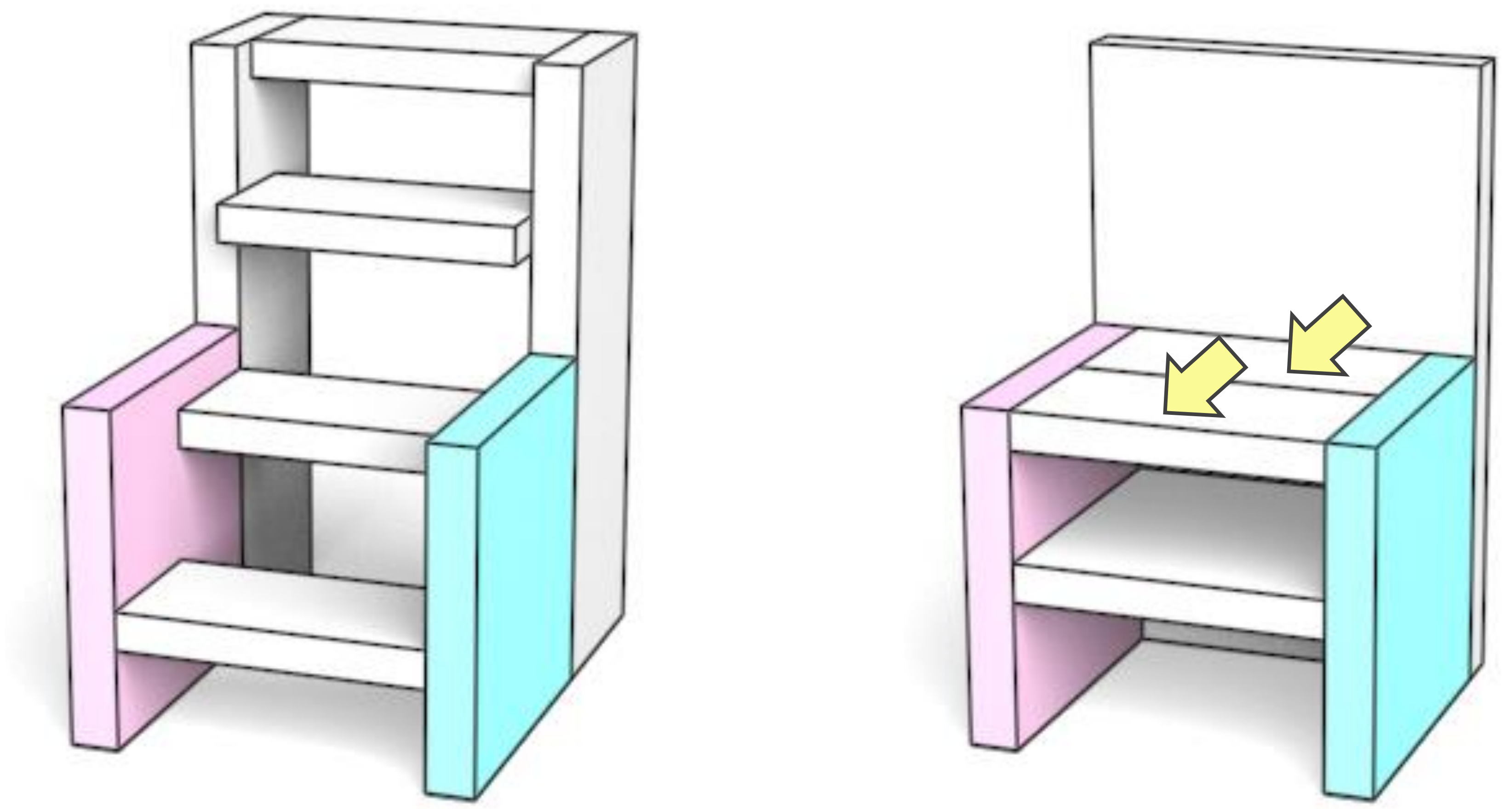
Resizing a large cage to match a small one will result in significant volume change on the cage.





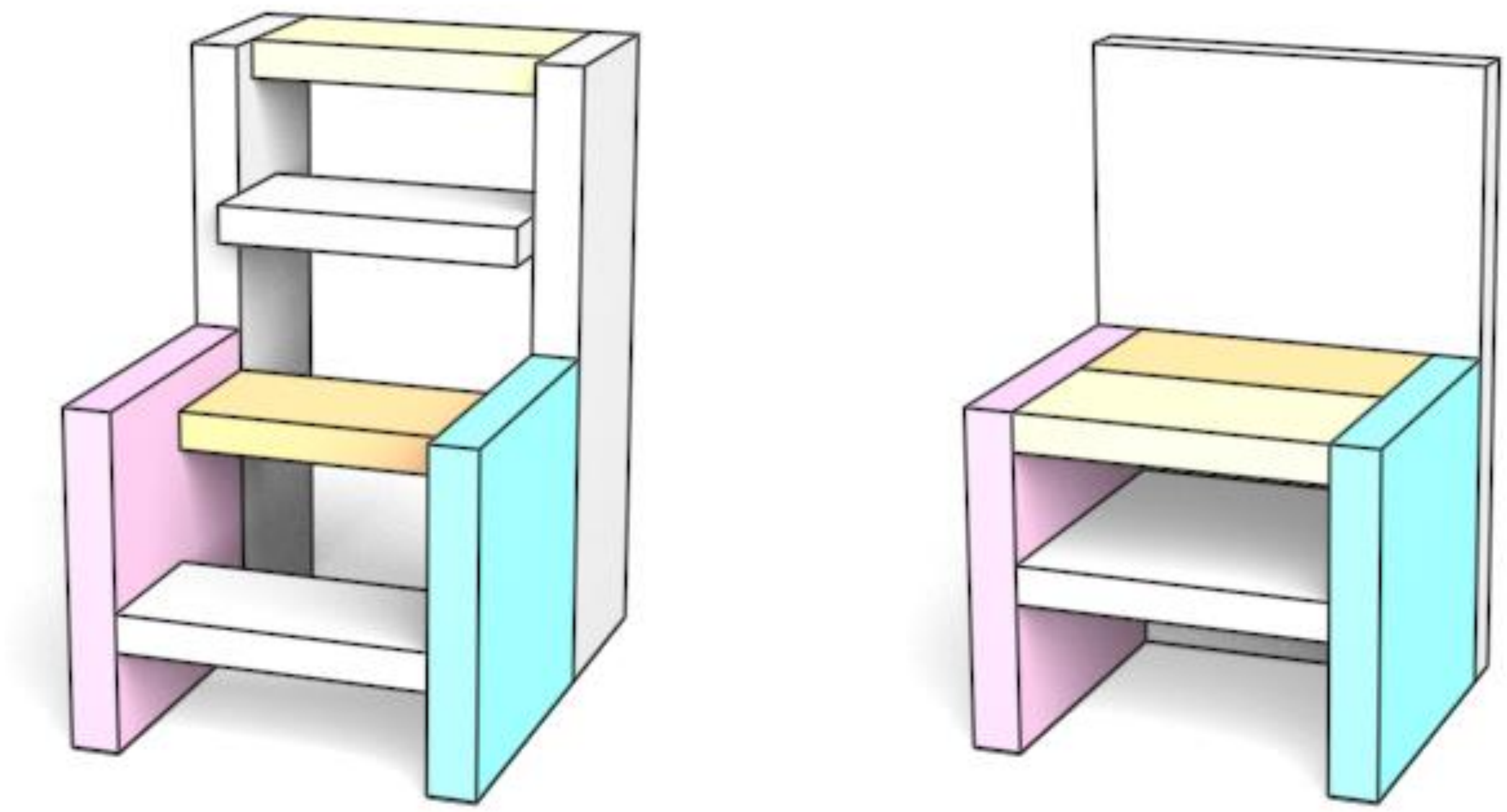
# **Operation #3:** Split Cages

Thus, we first split the large cage into two smaller cages



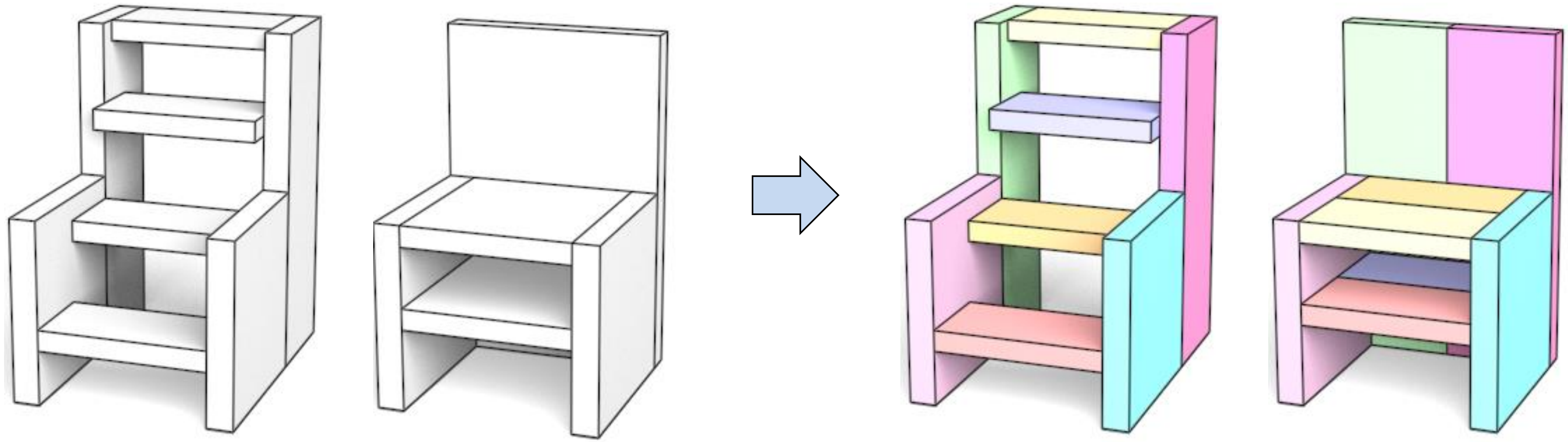
# **Operation #3:** Split Cages

and form a correspondence for each of the smaller cages.

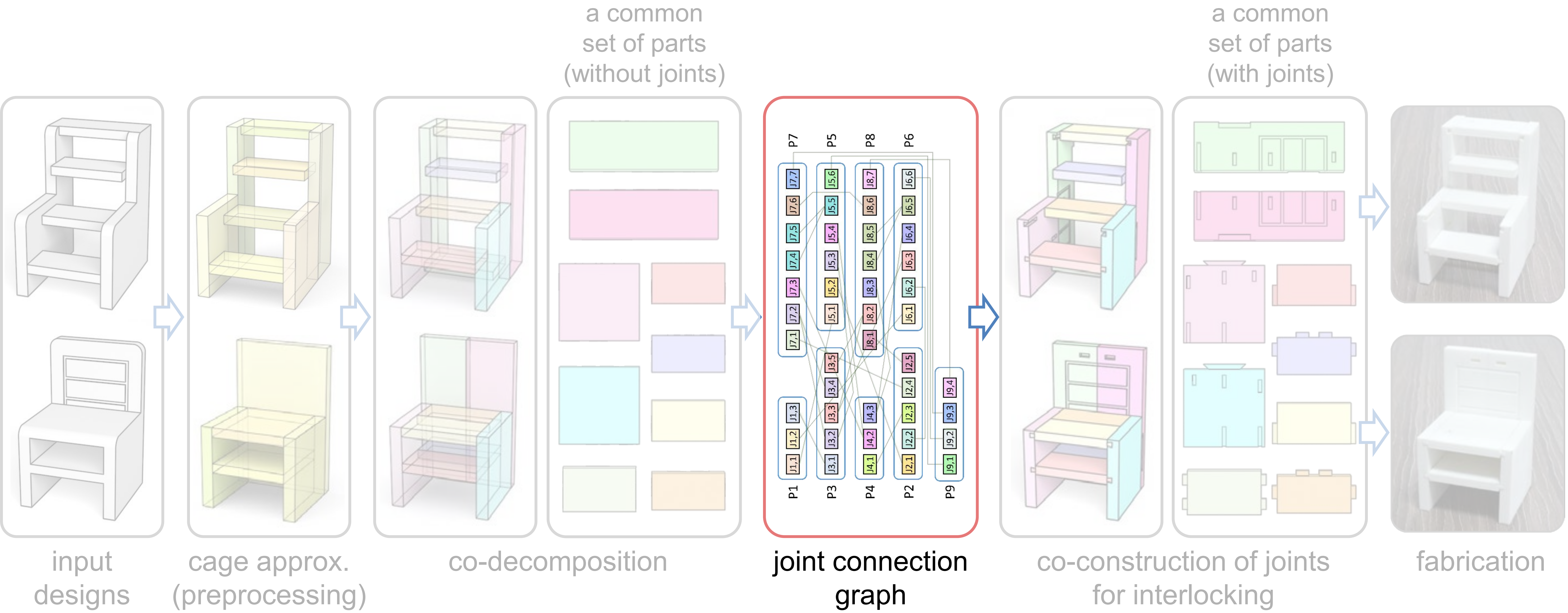


# Iterative Approach

- Develop an iterative approach that apply the operations with randomness
- Pick the best *K* results that maximize the objective function

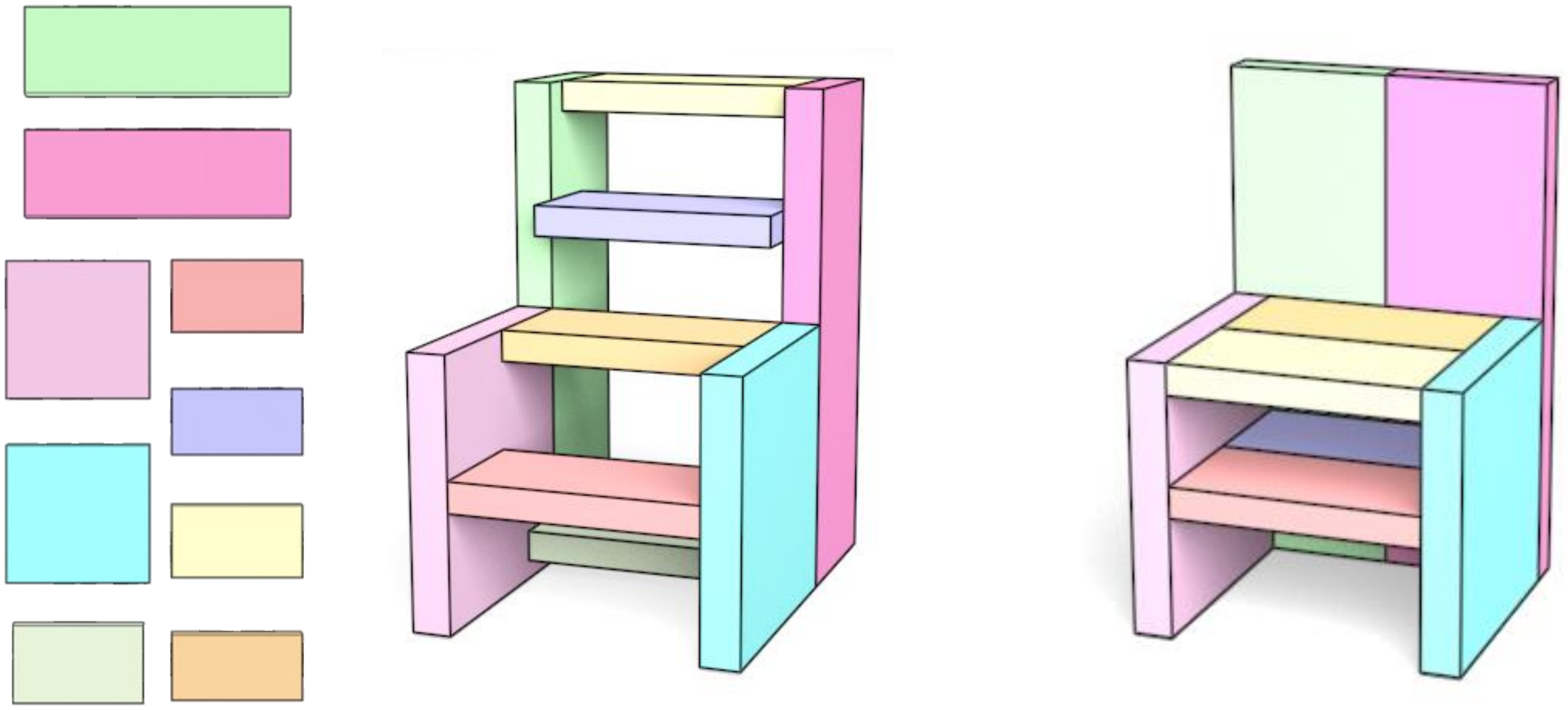


# Overview of Our Approach



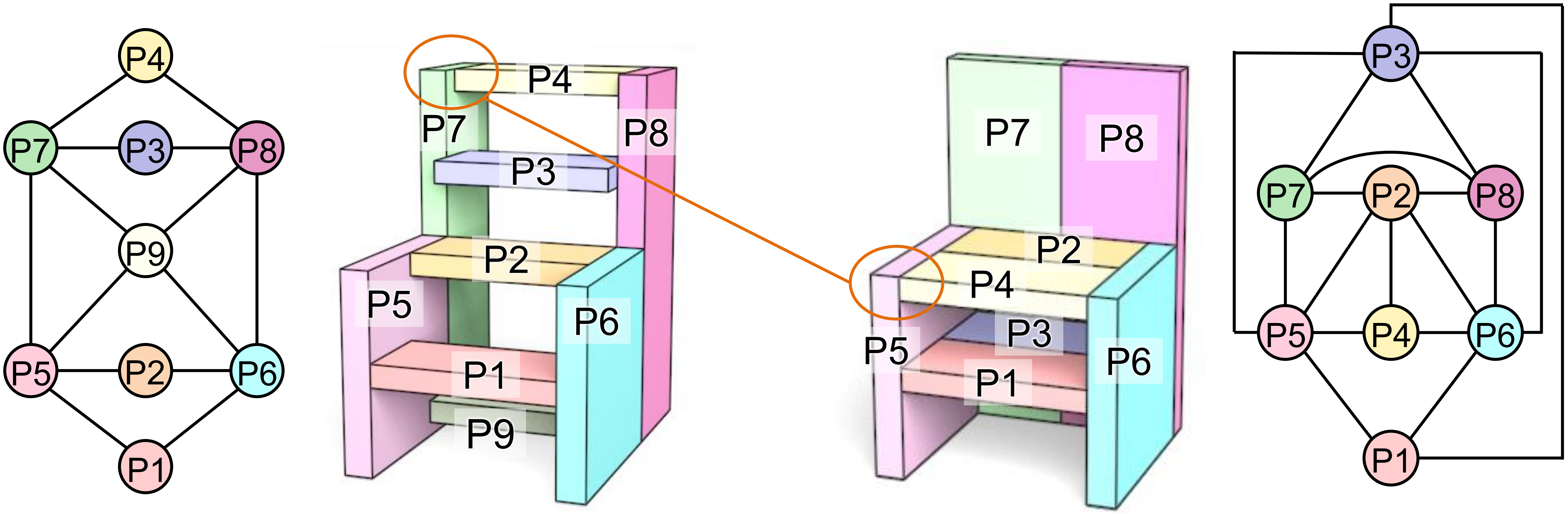
# Model Joint Connections

- The co-decomposition generates a common set of parts
- Yet, the parts have no joints for their physical connection



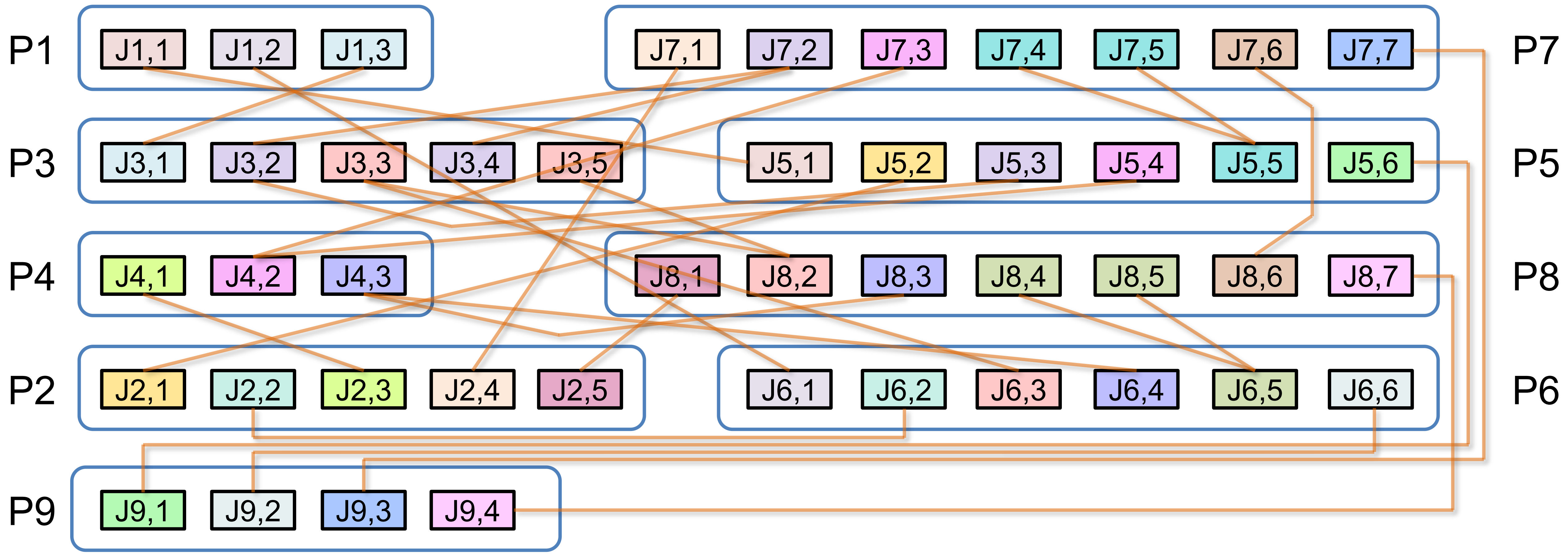
# Model Joint Connections

Parts-graphs model only joint connections in individual designs, yet not <u>compatible joint connections</u> across multiple designs



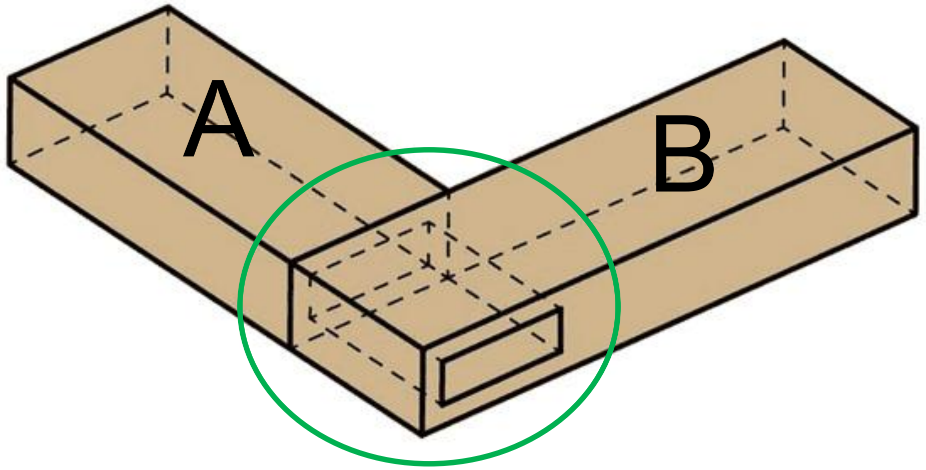
# Model Joint Connections

We develop a new data structure called **half-joint graph** to model these compatible joints connections.

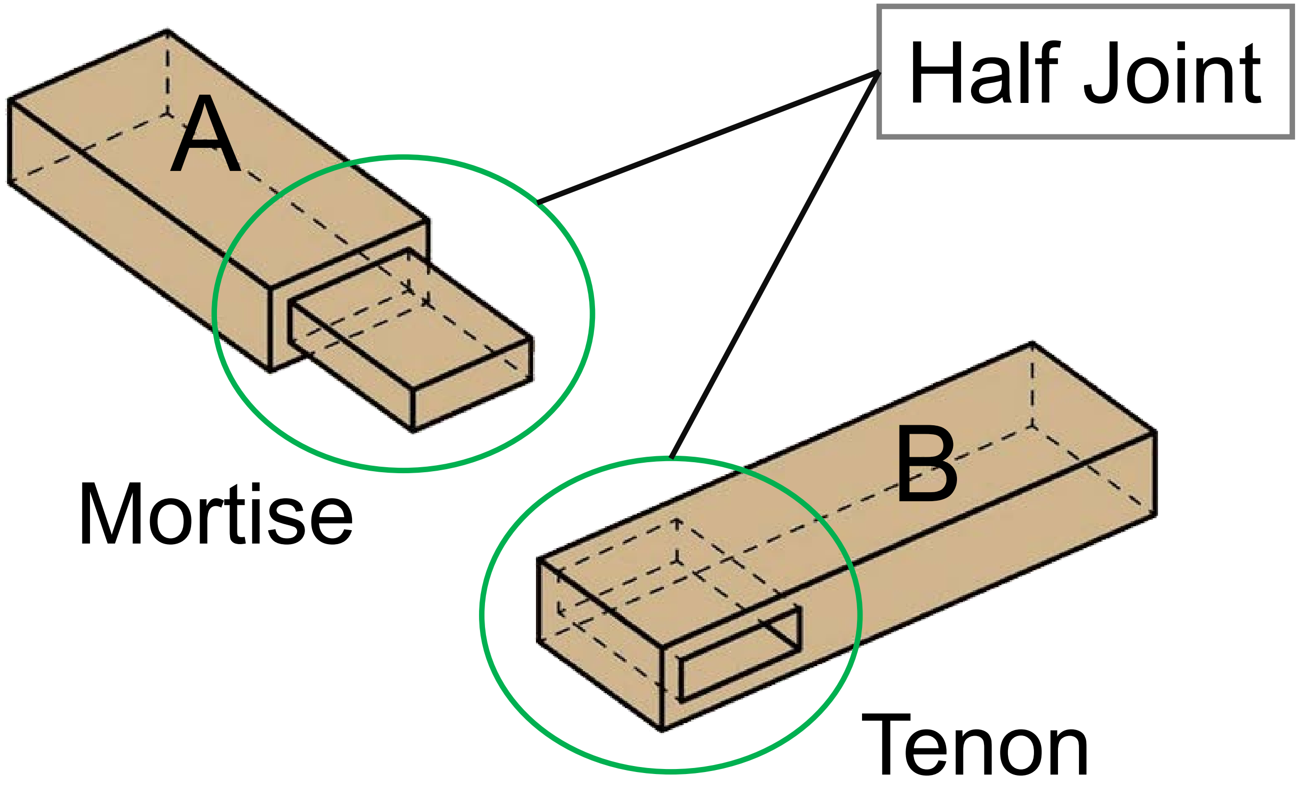


# Half Joint

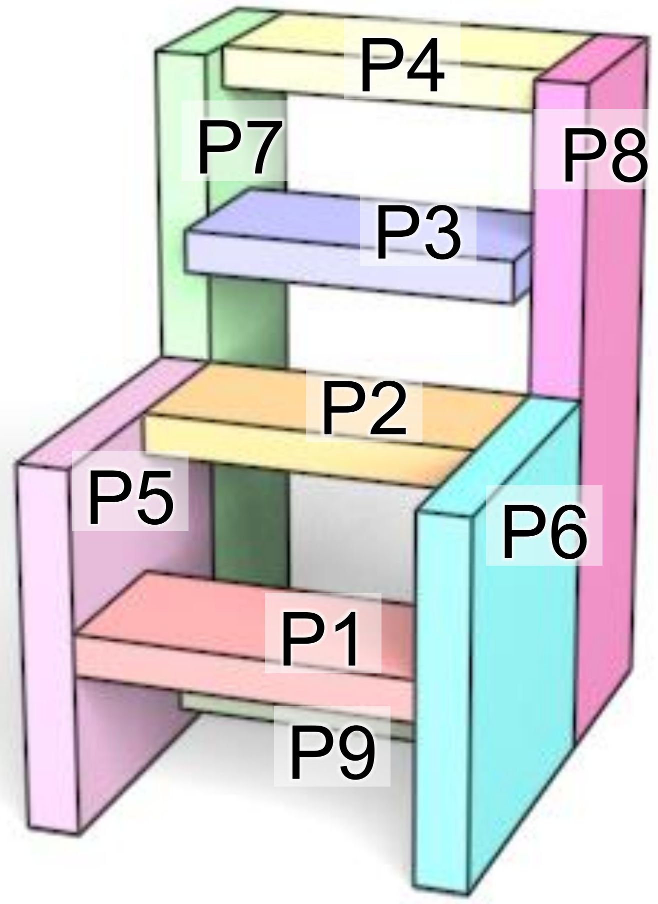
Half joint is the portion of a joint on an individual part

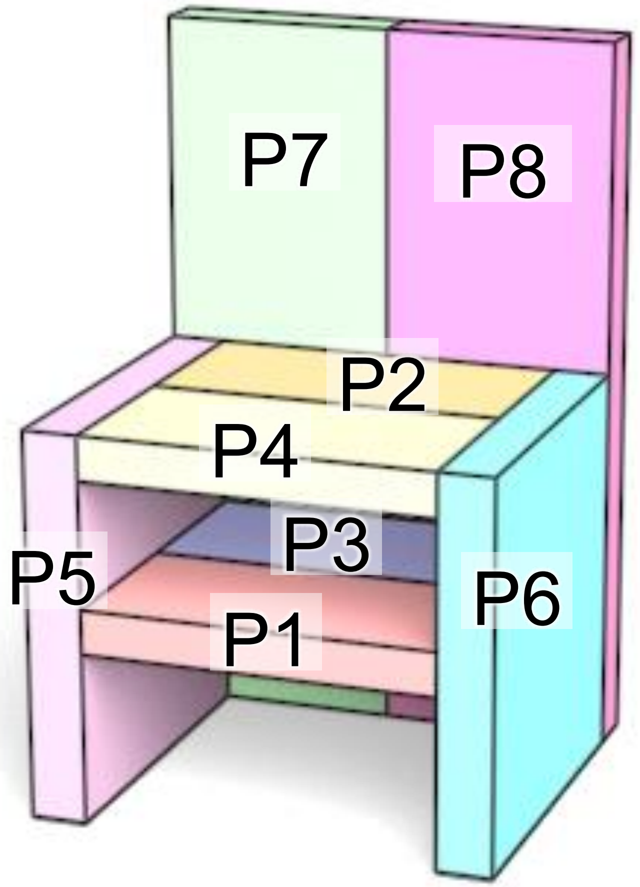



Mortise and tenon joint

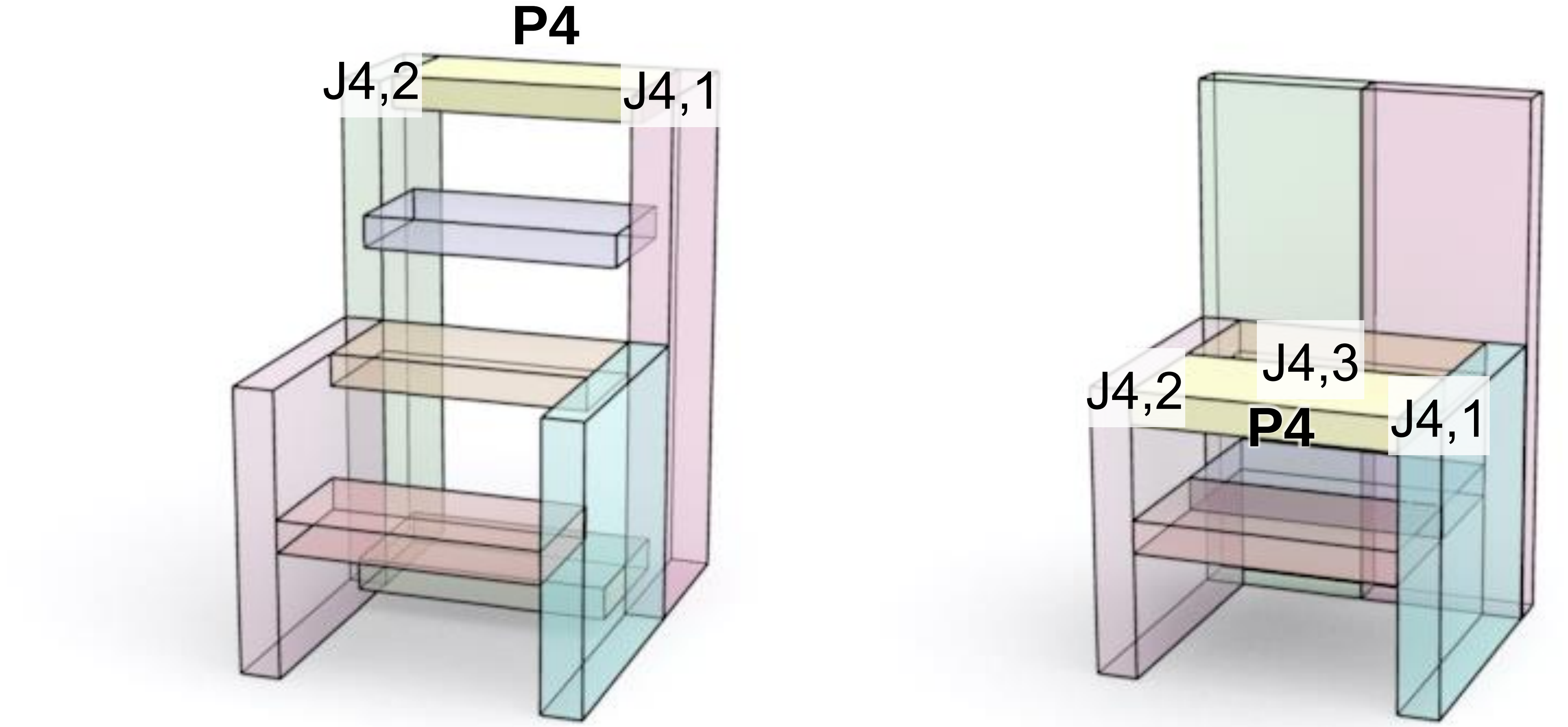


# Identify Half Joints

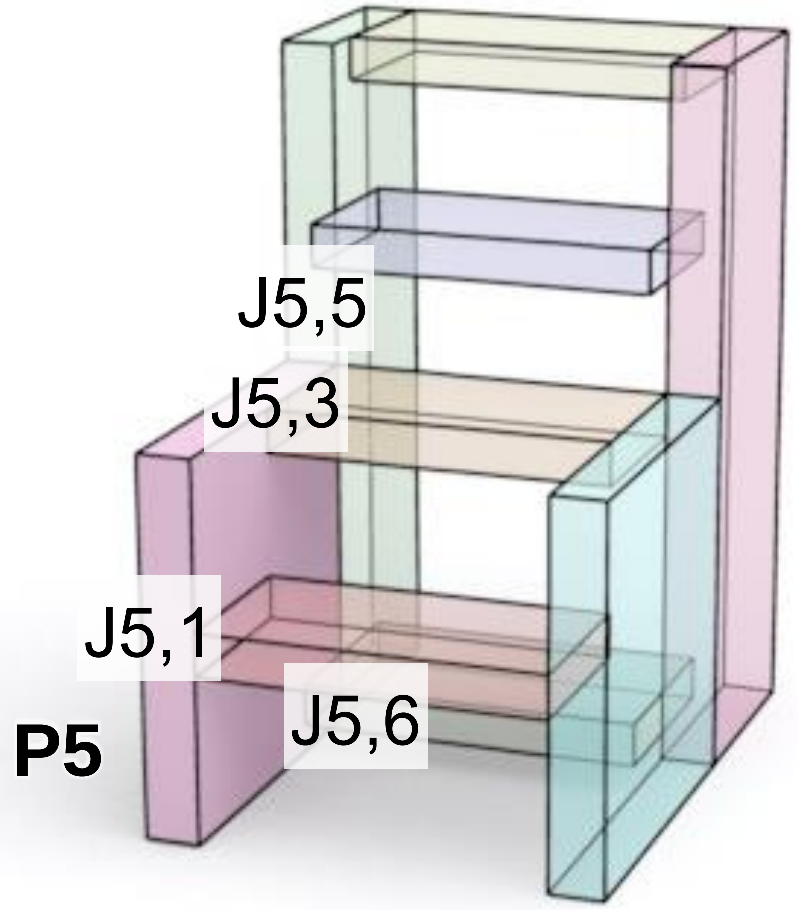


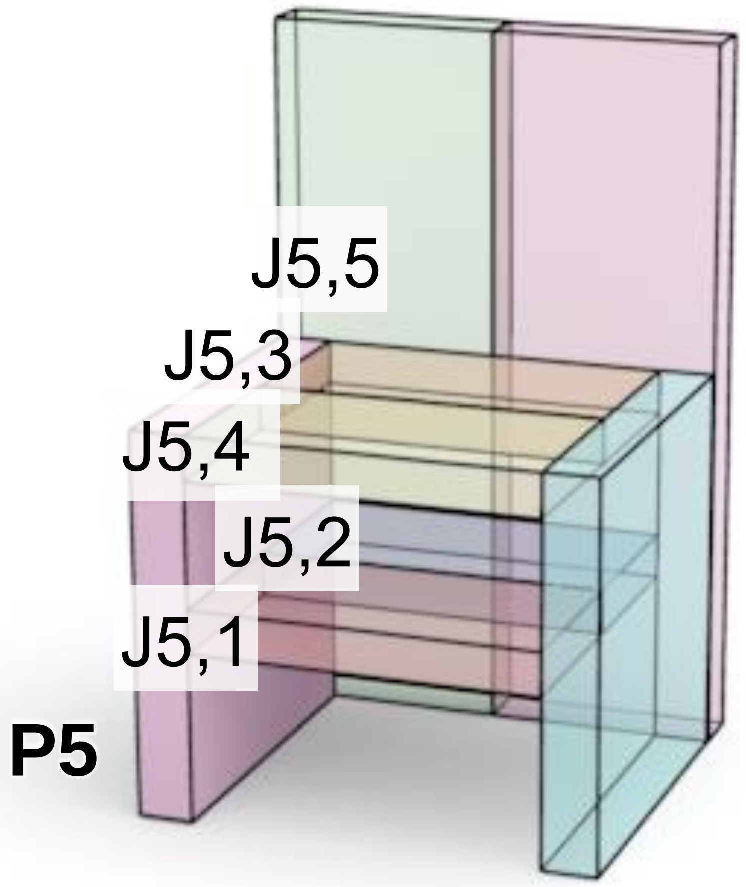



# Identify Half Joints

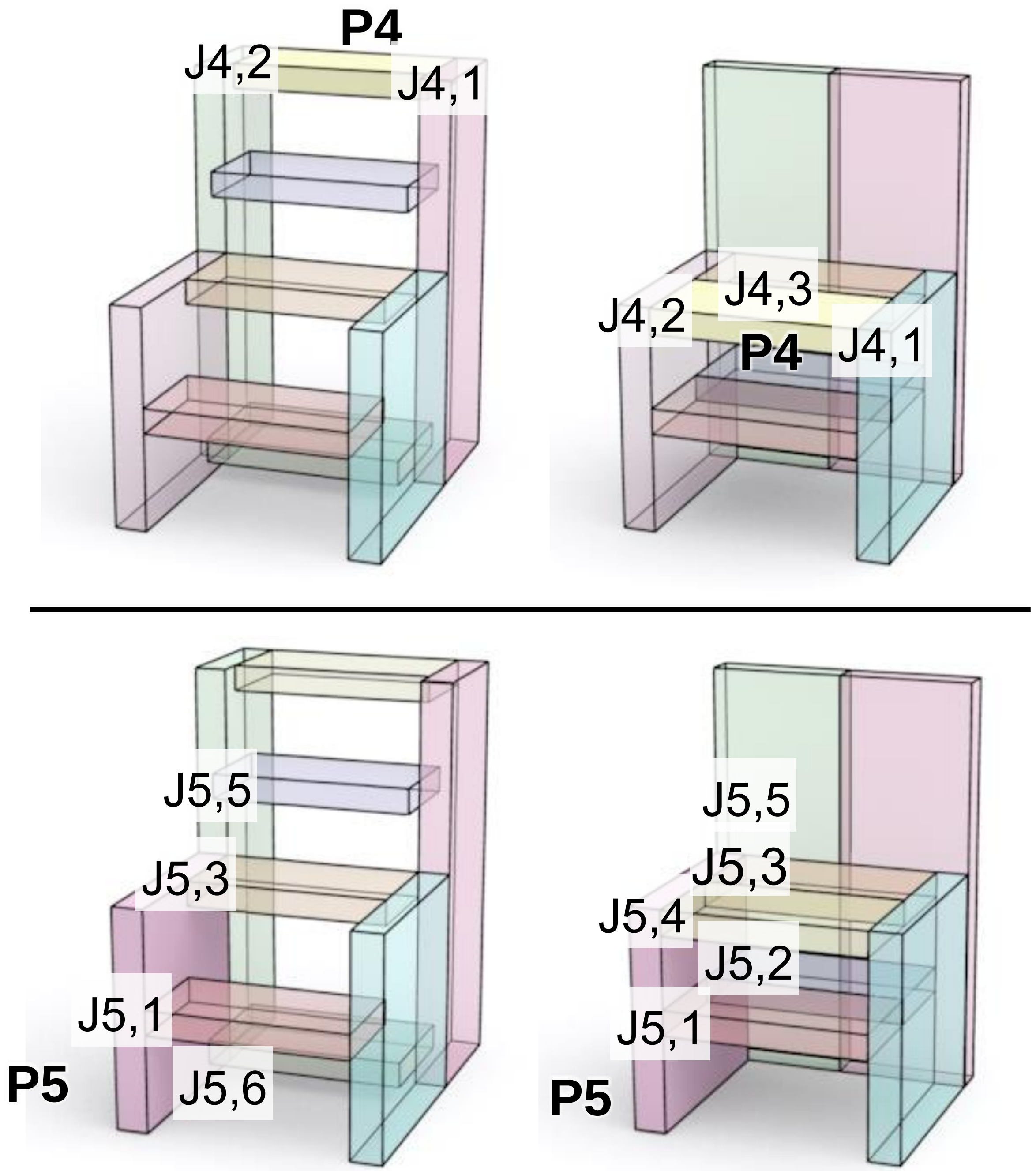


# Identify Half Joints



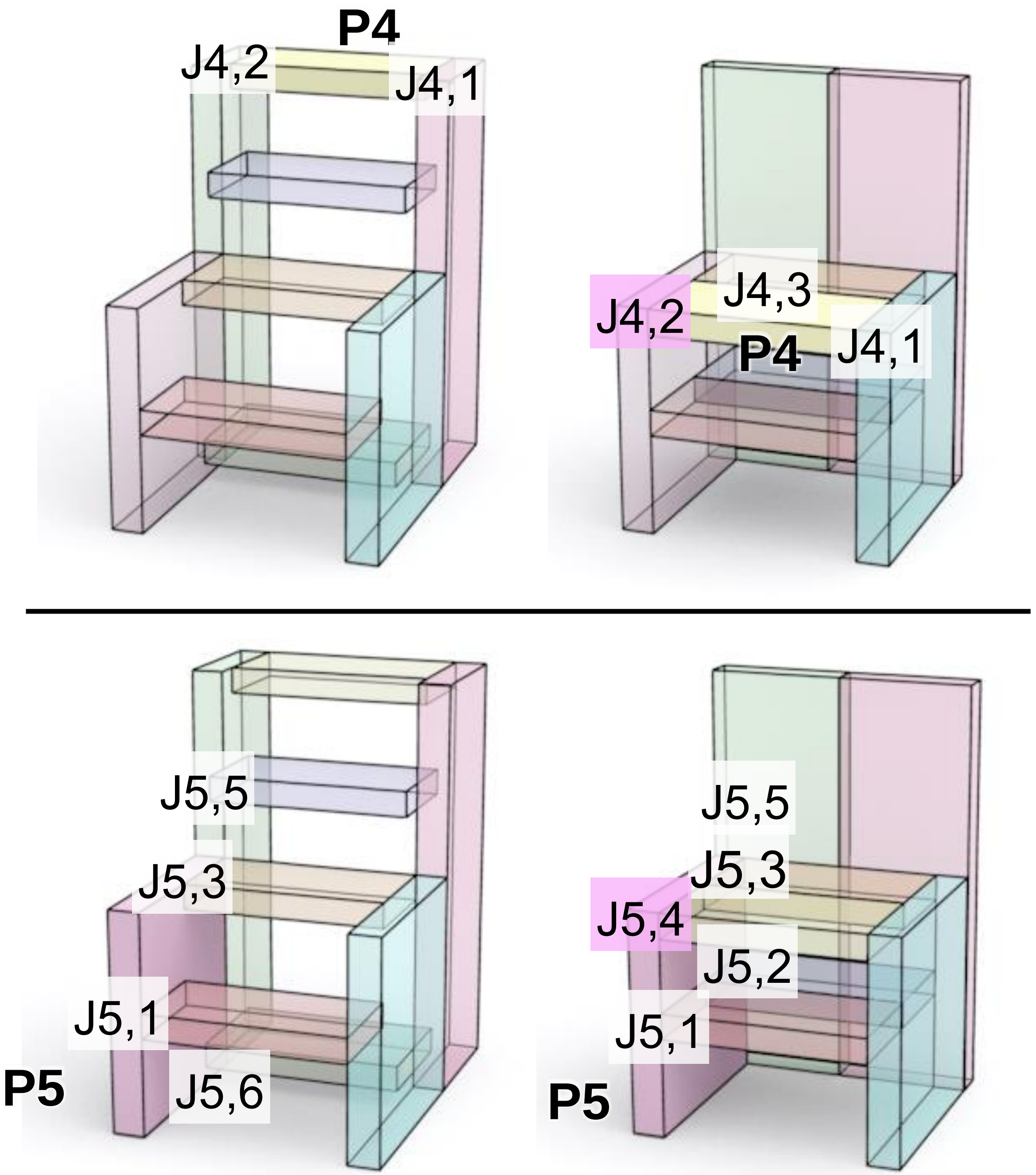
# Connect Half Joints

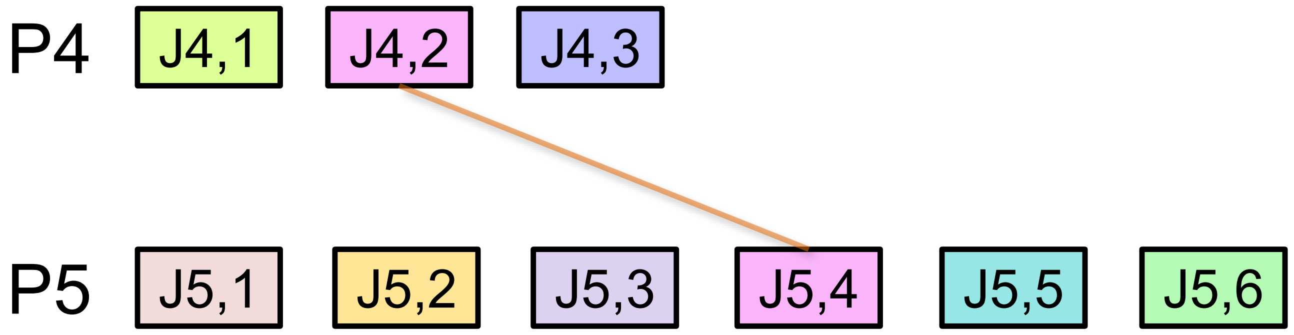



P4 J4,1 J4,2 J4,3

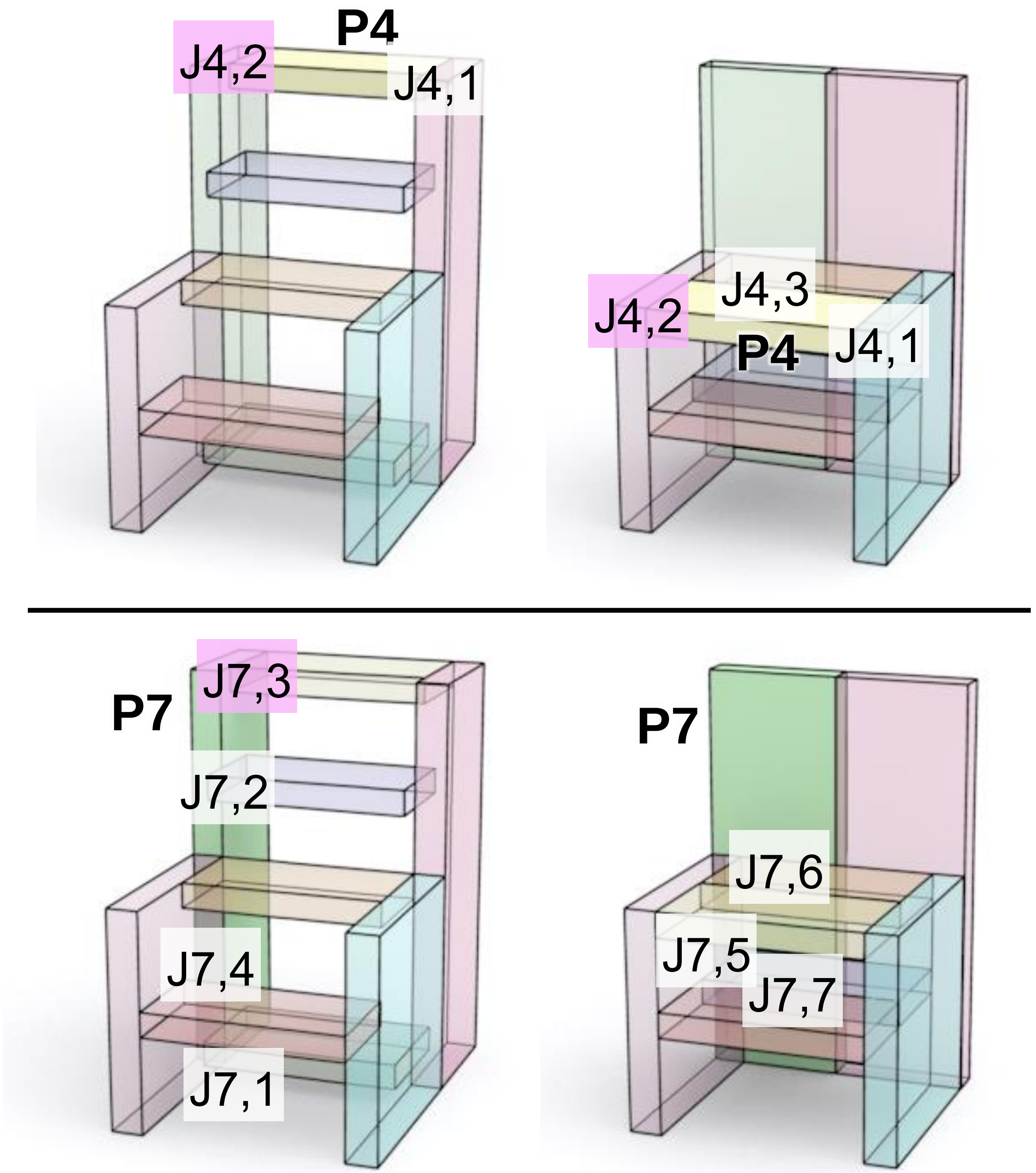
P5 J5,1 J5,2 J5,3 J5,4 J5,5 J5,6

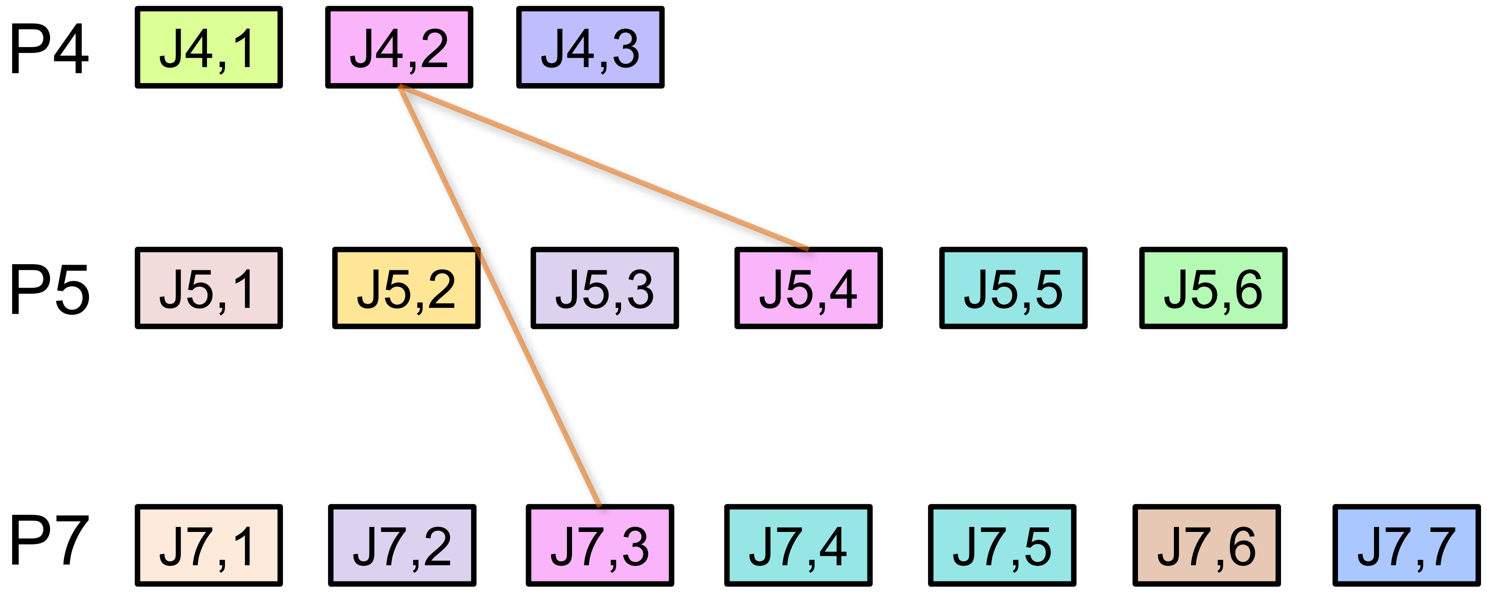
# Connect Half Joints



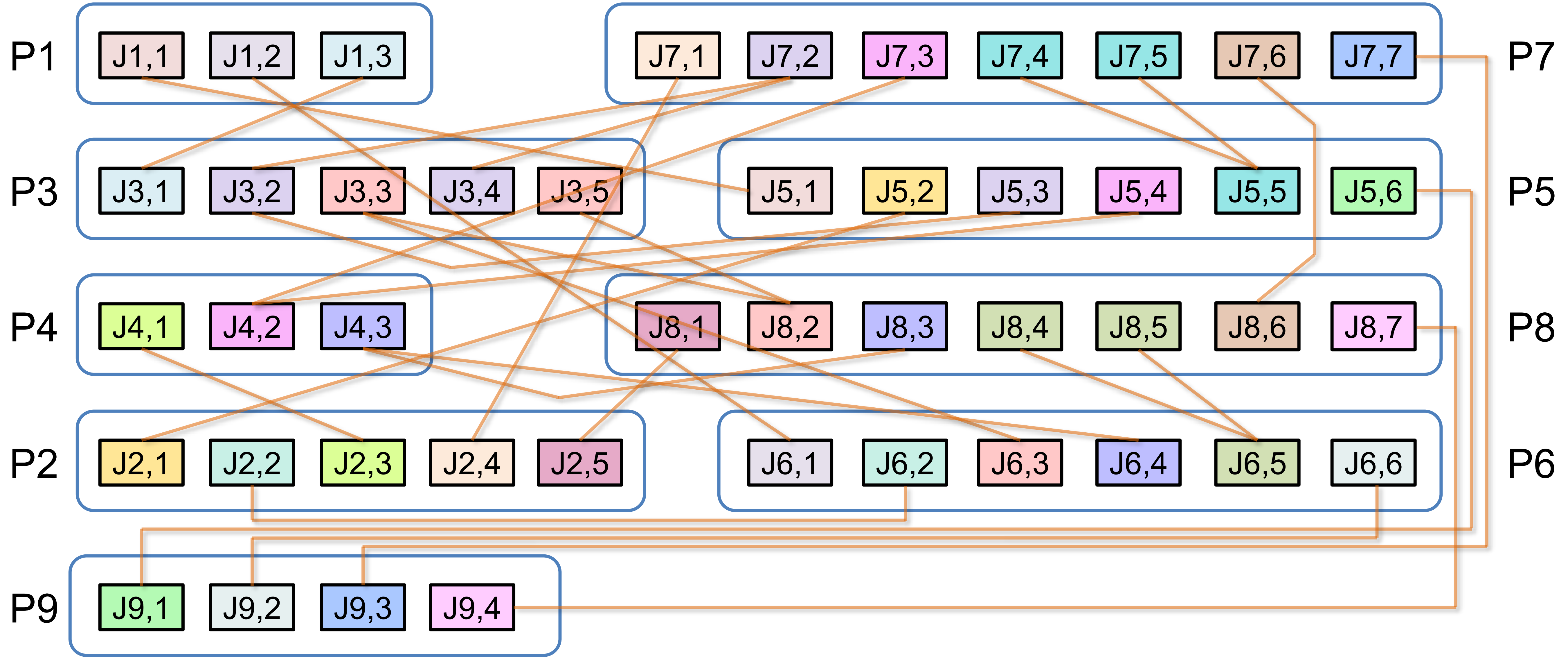



# Connect Half Joints



# Half-Joint Graph



# Formulate the Search Space

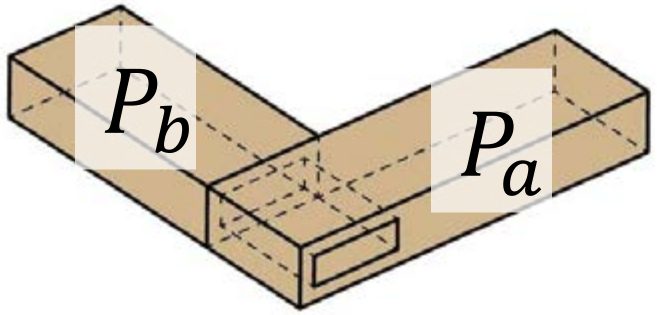
If half joints $J_{a,x}$ and $J_{b,y}$ connect parts $P_a$ and $P_b$ in the *k*-th design, we should have

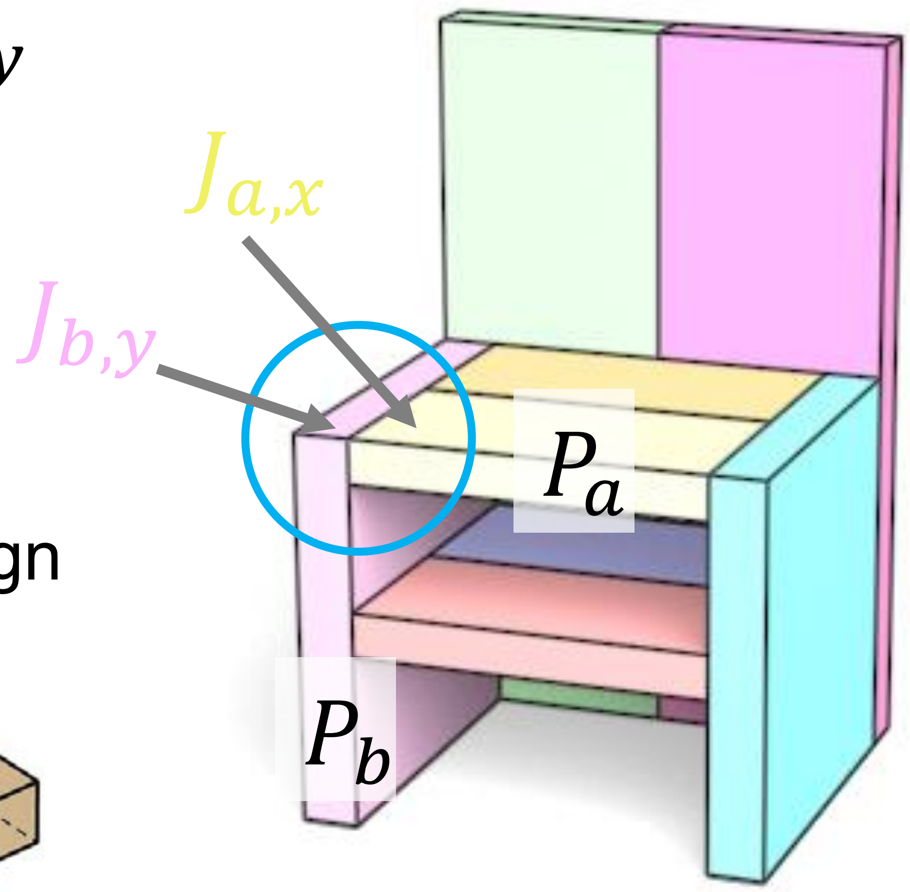
$$T_{k,a} \cdot d_{a,x} = -T_{k,b} \cdot d_{b,y}$$

$T_{k,a}$ is the transform of $P_a$ in k-th design

$d_{a,x}$ is the direction to remove $P_a$ from its partner at $J_{a,x}$

$T_{k,a} \cdot d_{a,x}$ is $P_a$'s removal direction at $J_{a,x}$ in the *k*-th design

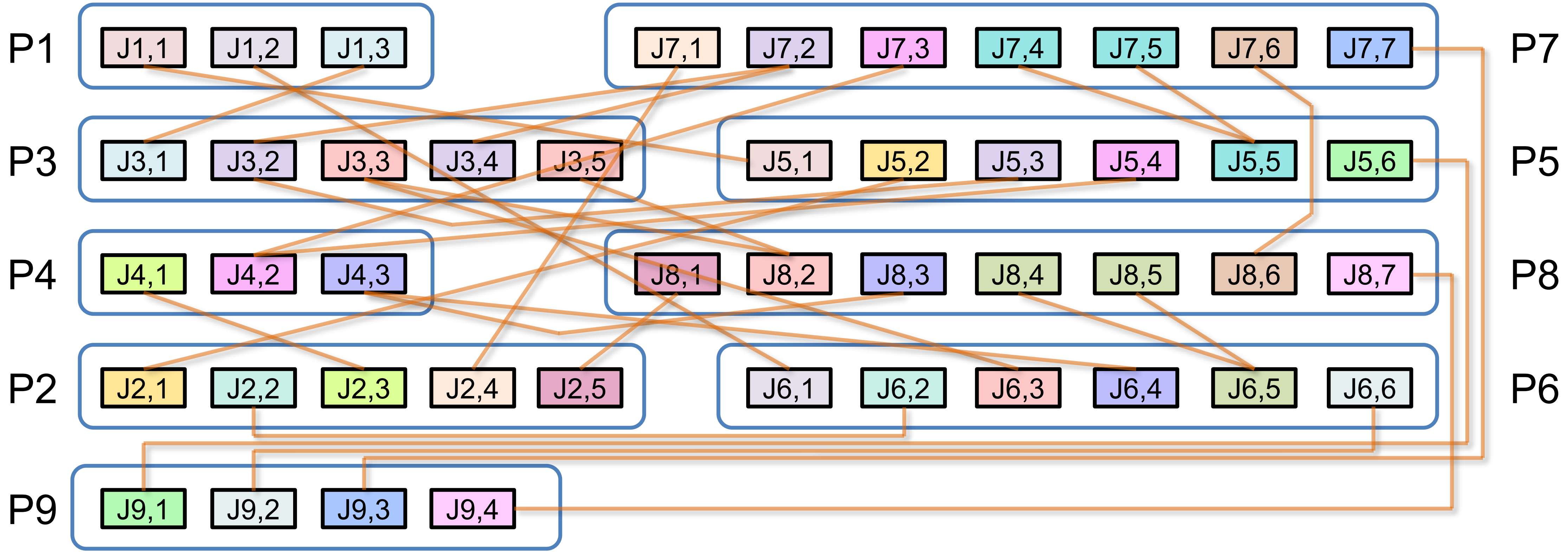



K-th design

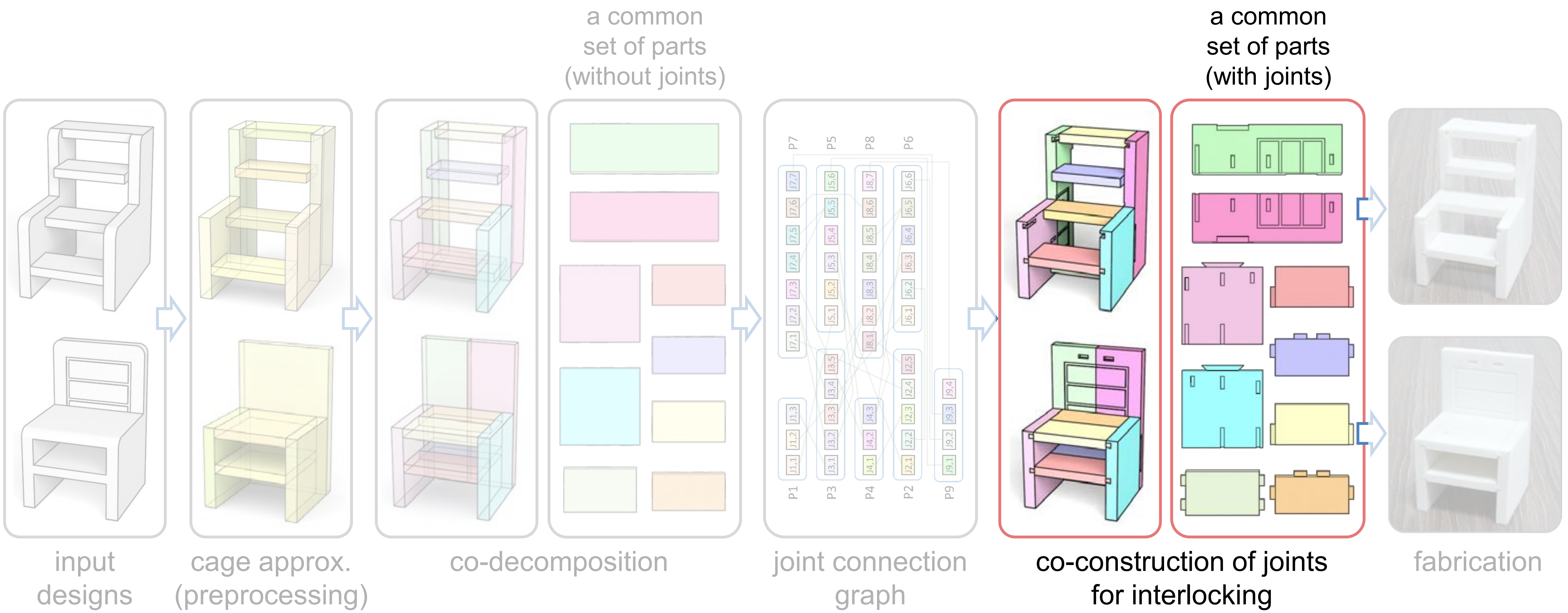
# Formulate the Search Space

The actual number of free variables depends on the number of <u>disjoint subgraphs</u> in the half-joint graph.



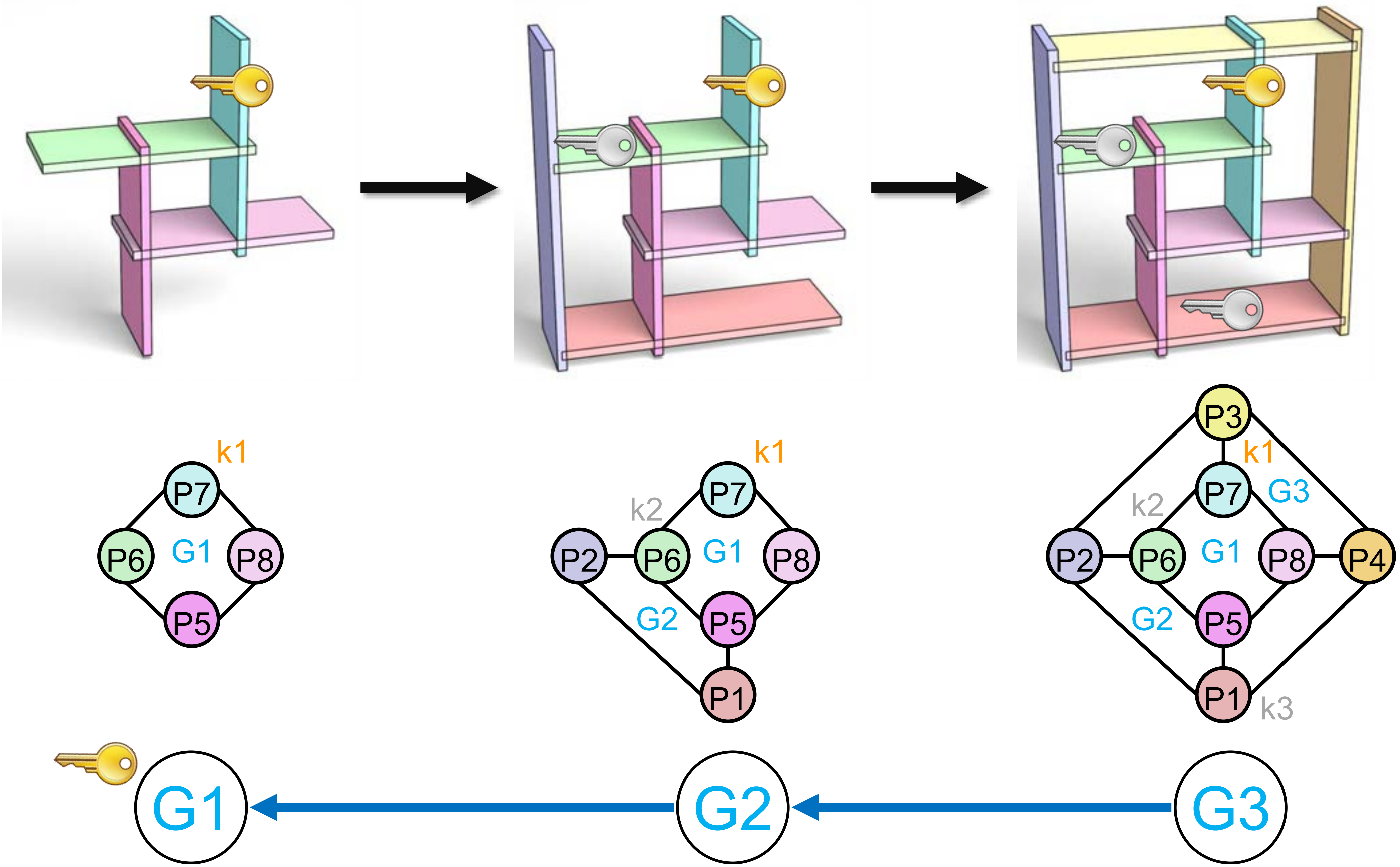

19 subgraphs

# Overview of Our Approach

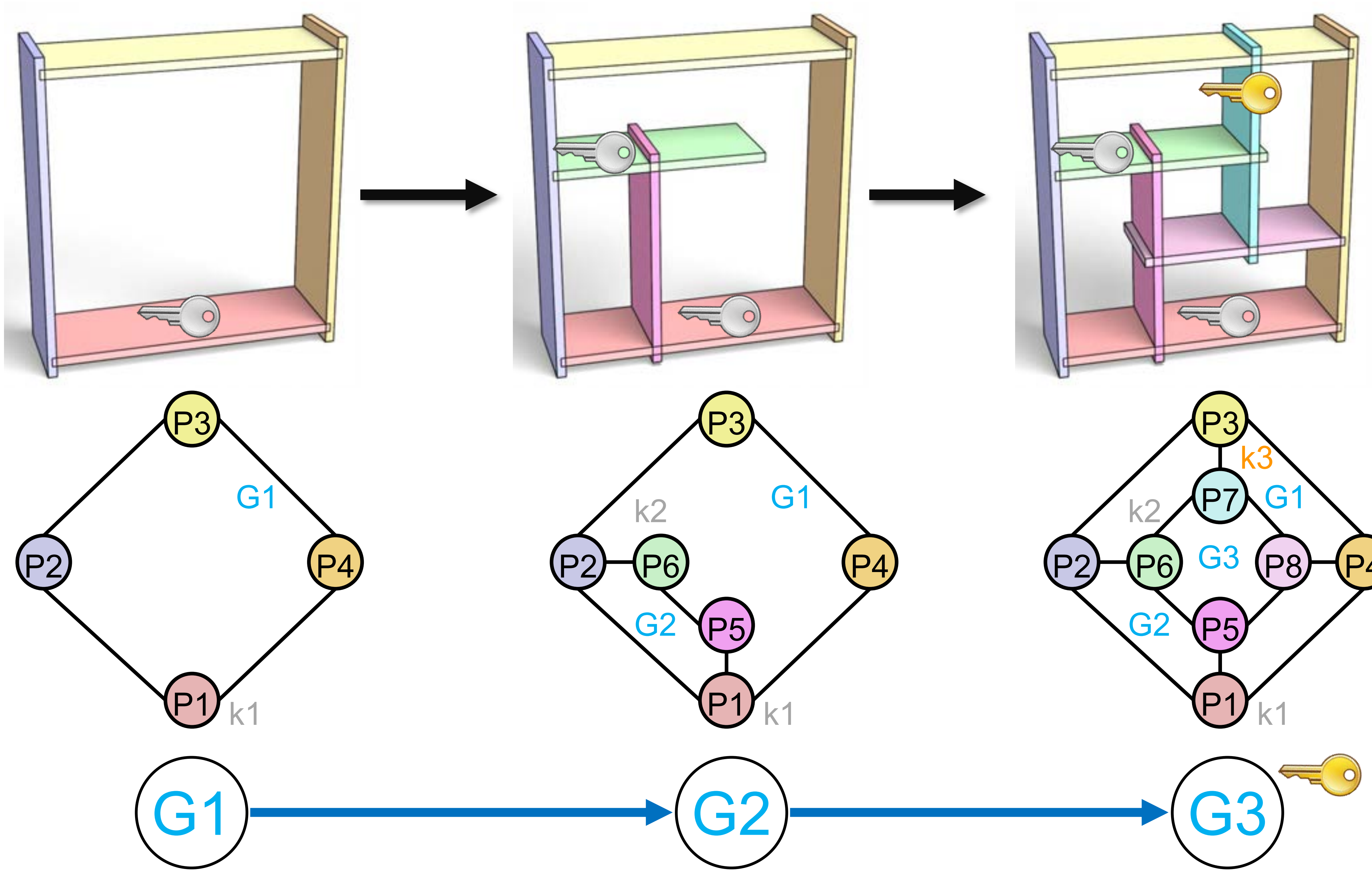


# Forward Interlocking

[Our previous work in SIG 2015]



# Backward Interlocking

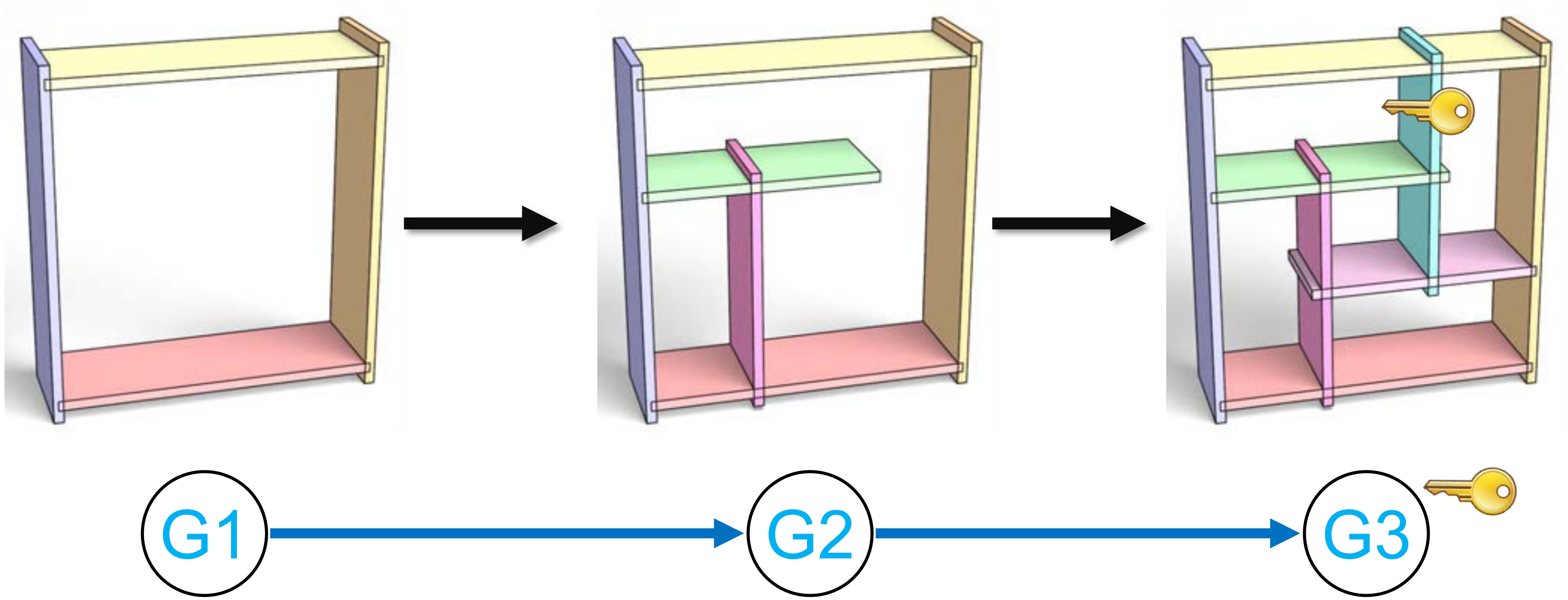


# Backward vs Forward Interlocking

Backward interlocking allows to

1) first explore substructures common to all designs

2) construct different global key(s) for different designs



# Multi-key Interlocking

Only a few specific keys are mobile; all parts and all subsets of parts are immobilized when these keys are held in place.

# Multi-key Interlocking

Only a few specific keys are mobile; all parts and all subsets of parts are immobilized when these keys are held in place.

# Multi-key Interlocking

Only a few specific keys are mobile; all parts and all subsets of parts are immobilized when these keys are held in place.

# Multi-key Interlocking

Only a few specific keys are mobile; all parts and all subsets of parts are immobilized when these keys are held in place.

# Multi-key Interlocking

Only a few specific keys are mobile; all parts and all subsets of parts are immobilized when these keys are held in place.

# Multi-key Interlocking

Only a few specific keys are mobile; all parts and all subsets of parts are immobilized when these keys are held in place.

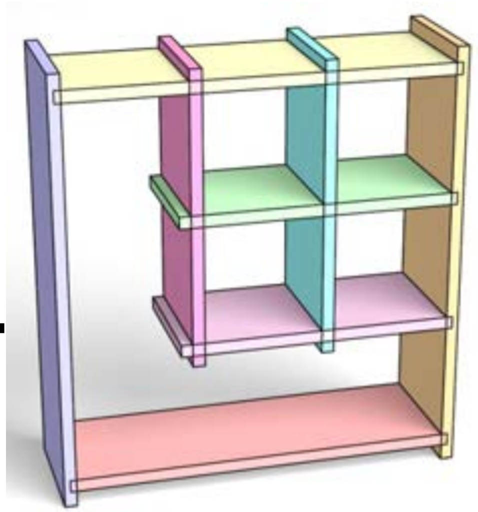
# Multi-key Interlocking

Only a few specific keys are mobile; all parts and all subsets of parts are immobilized when these keys are held in place.

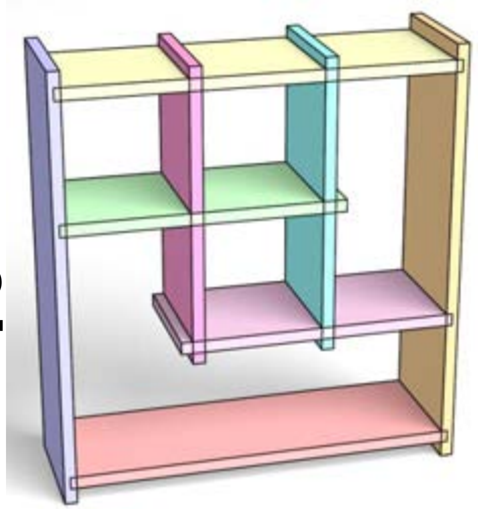
# Co-Construct Joints

Find common substructures of a given set of designs

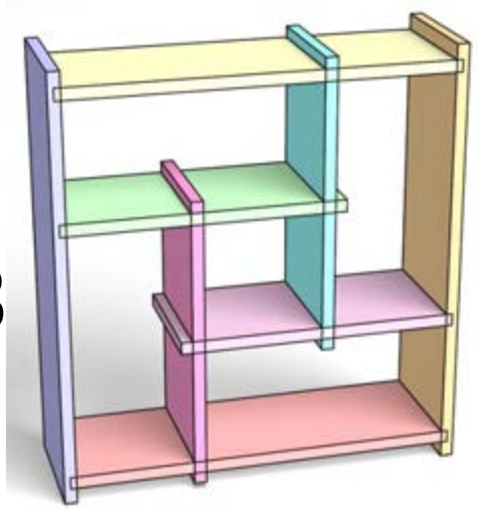
D1



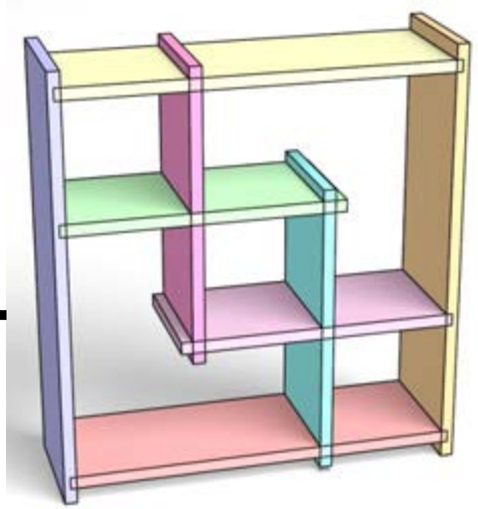
D2



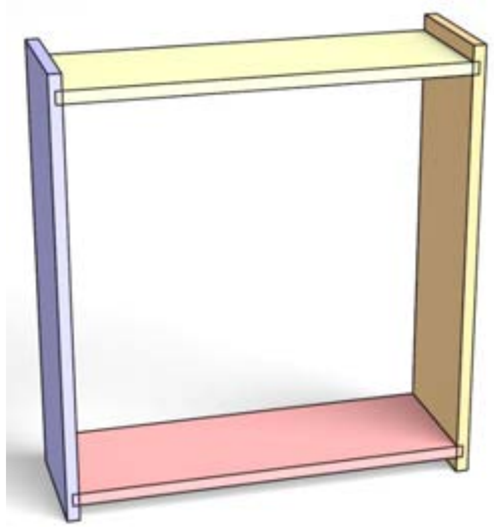
D3



D4



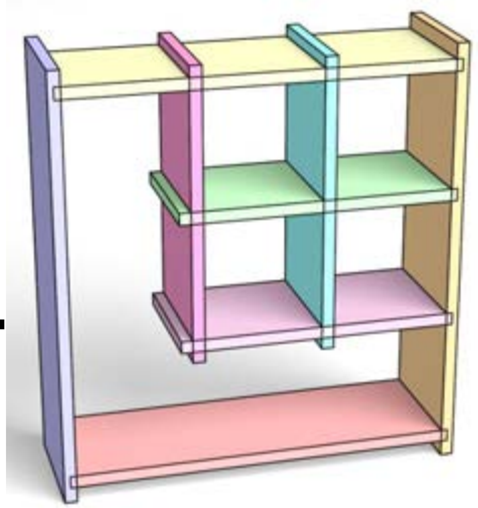
G1
[D1, D2, D3, D4]



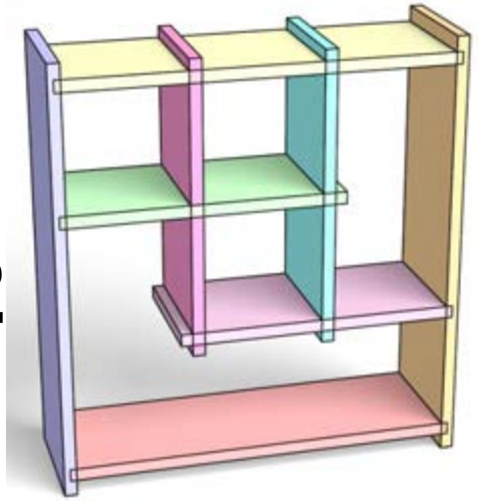
# Co-Construct Joints

Find common substructures of a given set of designs

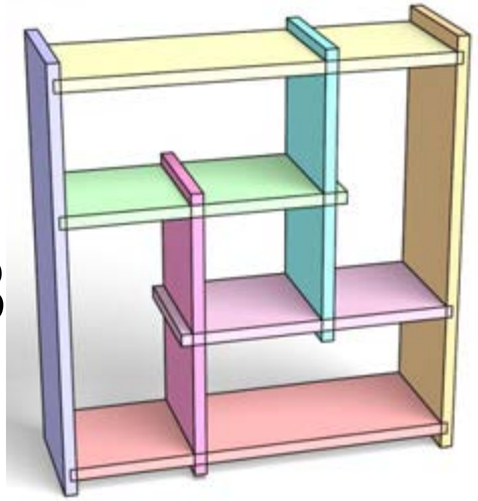
**D1**



**D2**



**D3**

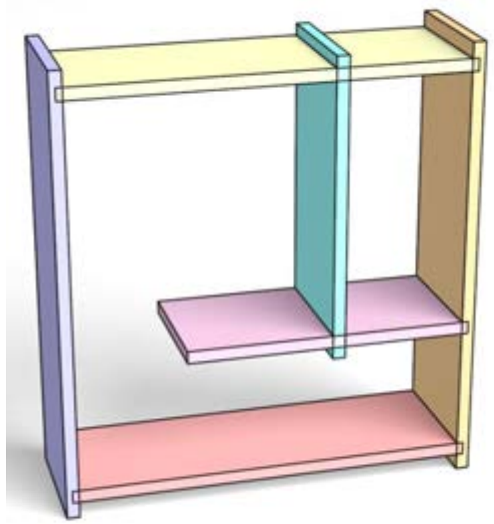


D4



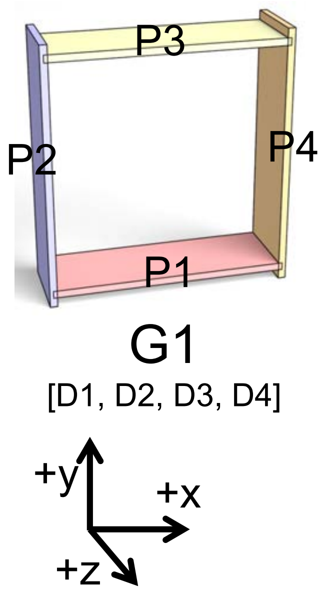
G1

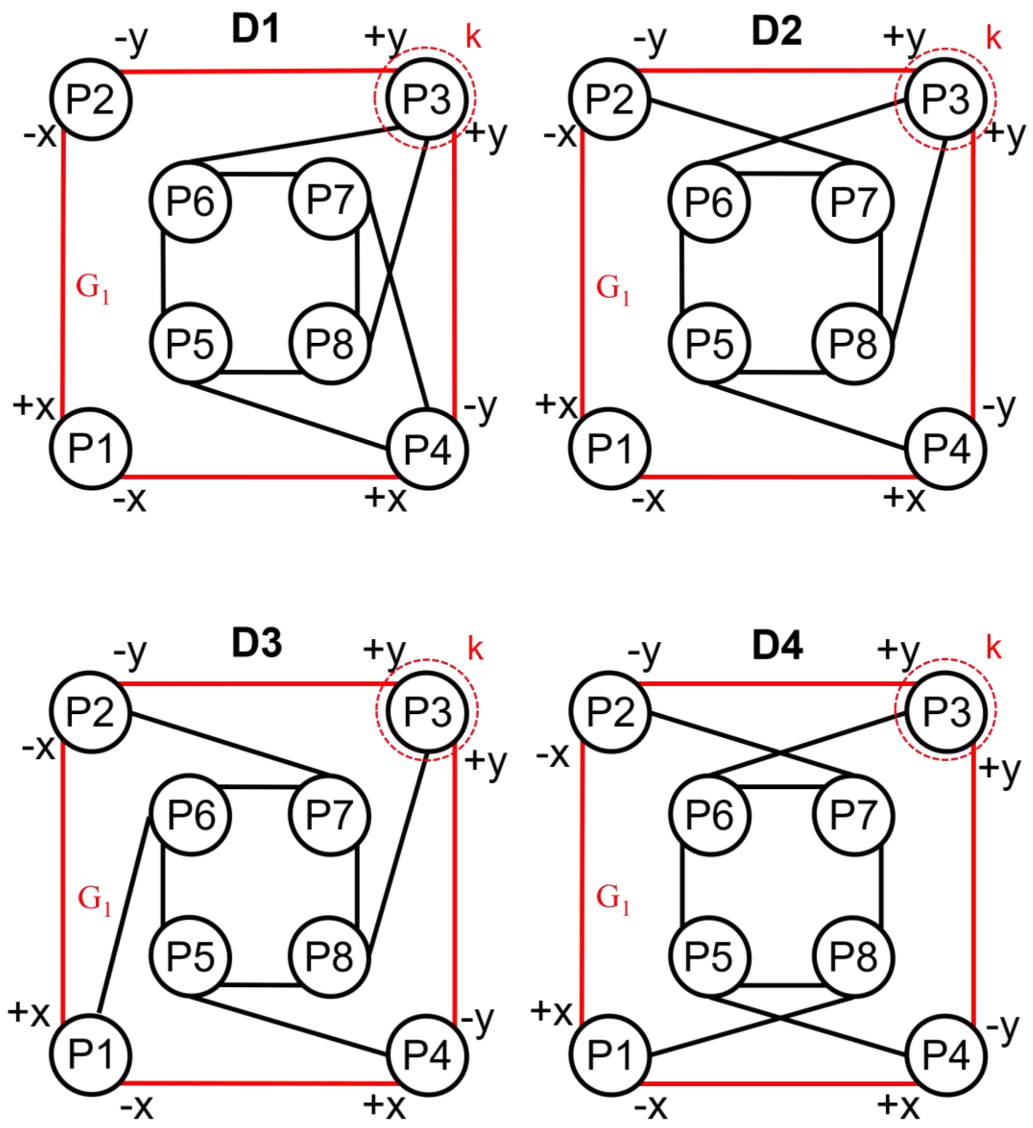
[D1, D2, D3]



# Co-Construct Joints

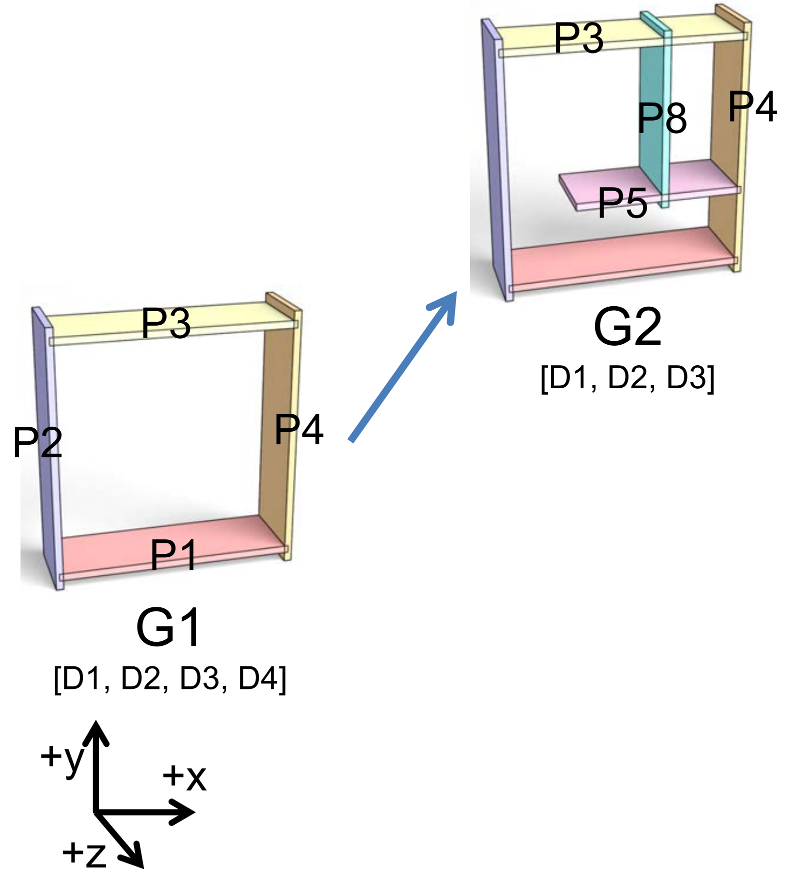
Plan joints iteratively

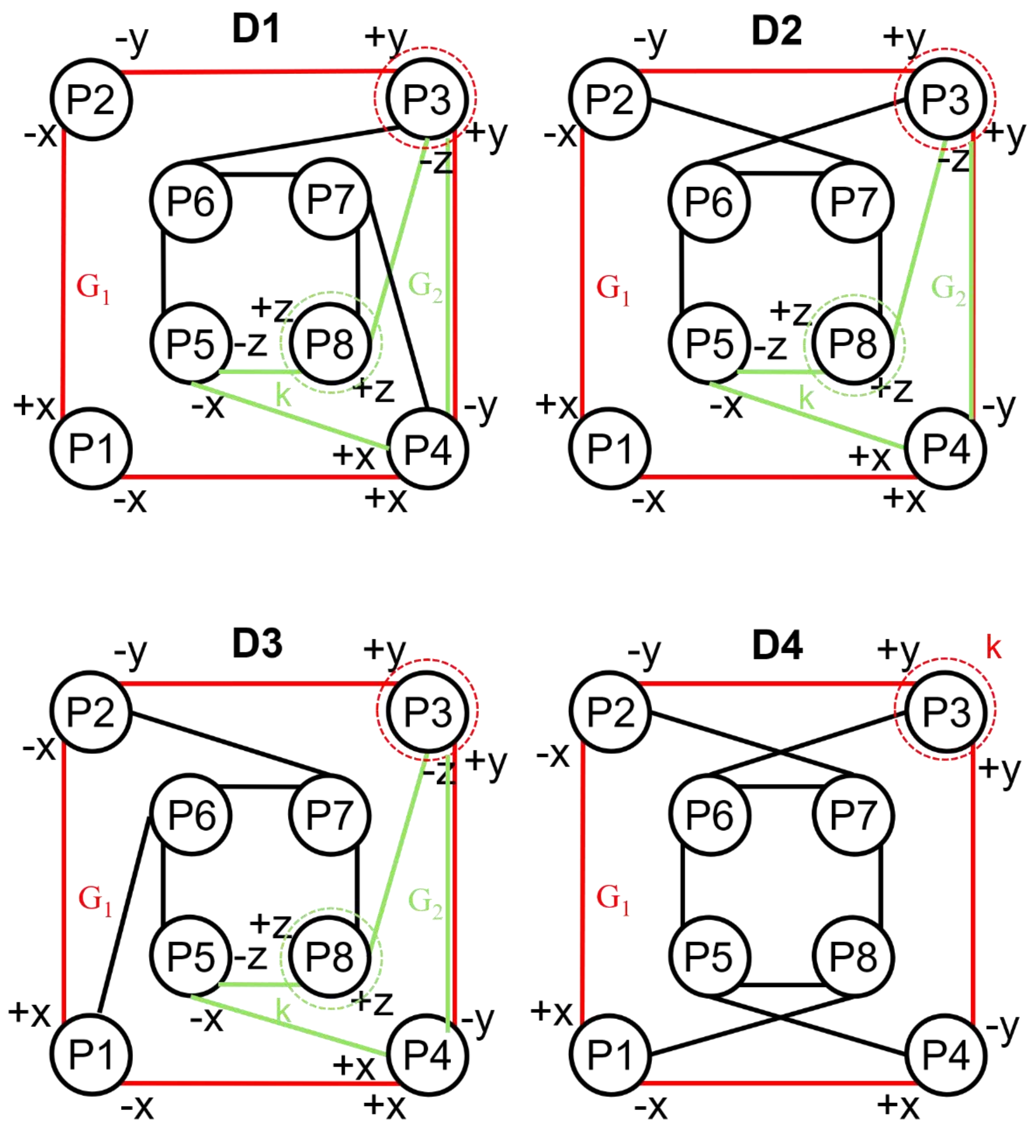


# Co-Construct Joints

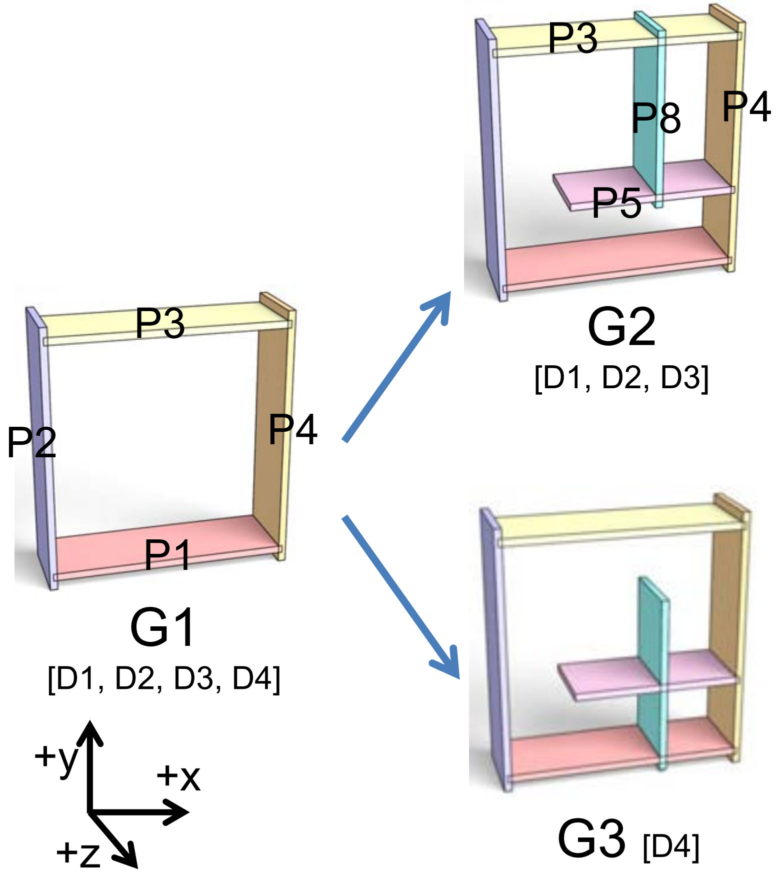
Plan joints iteratively

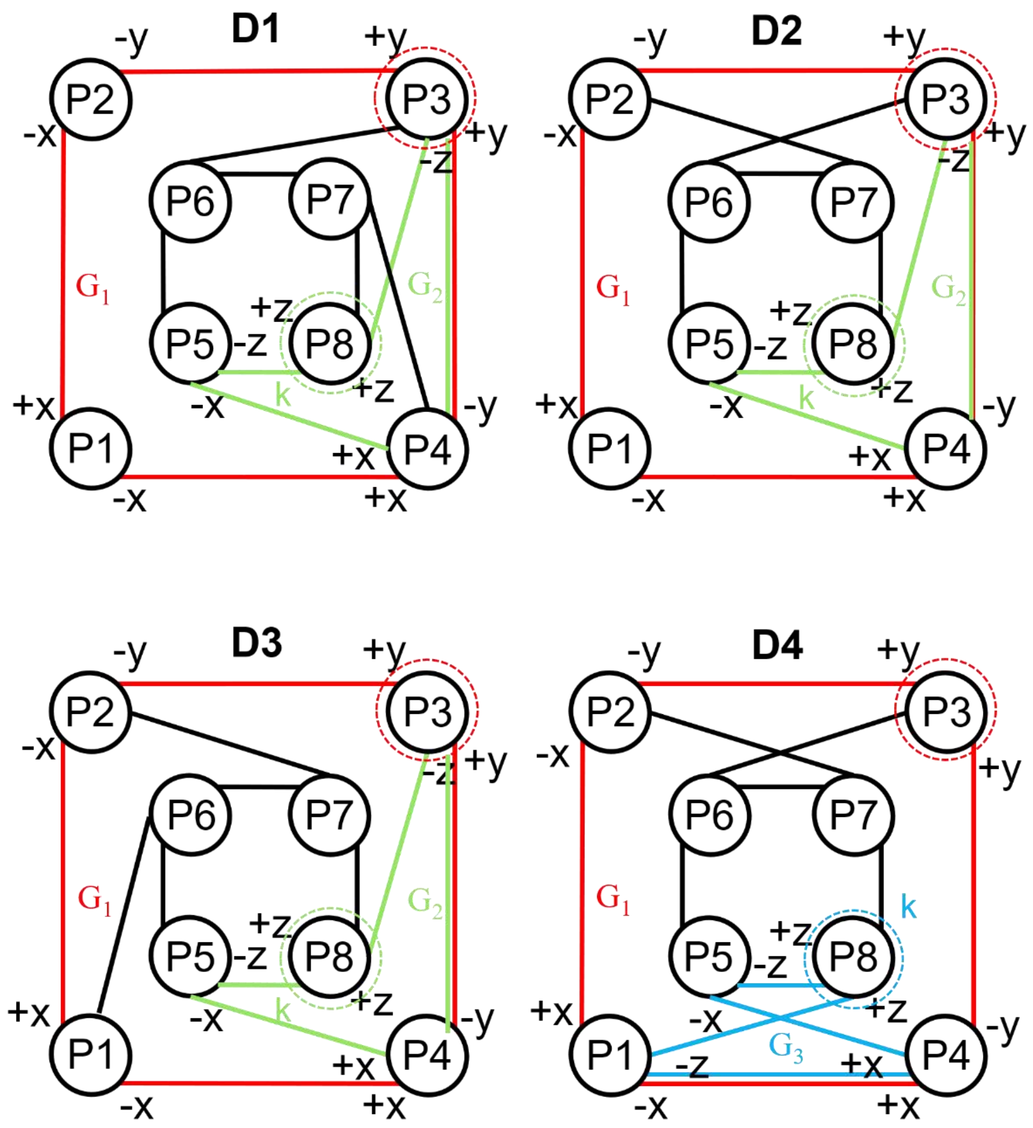


# Co-Construct Joints

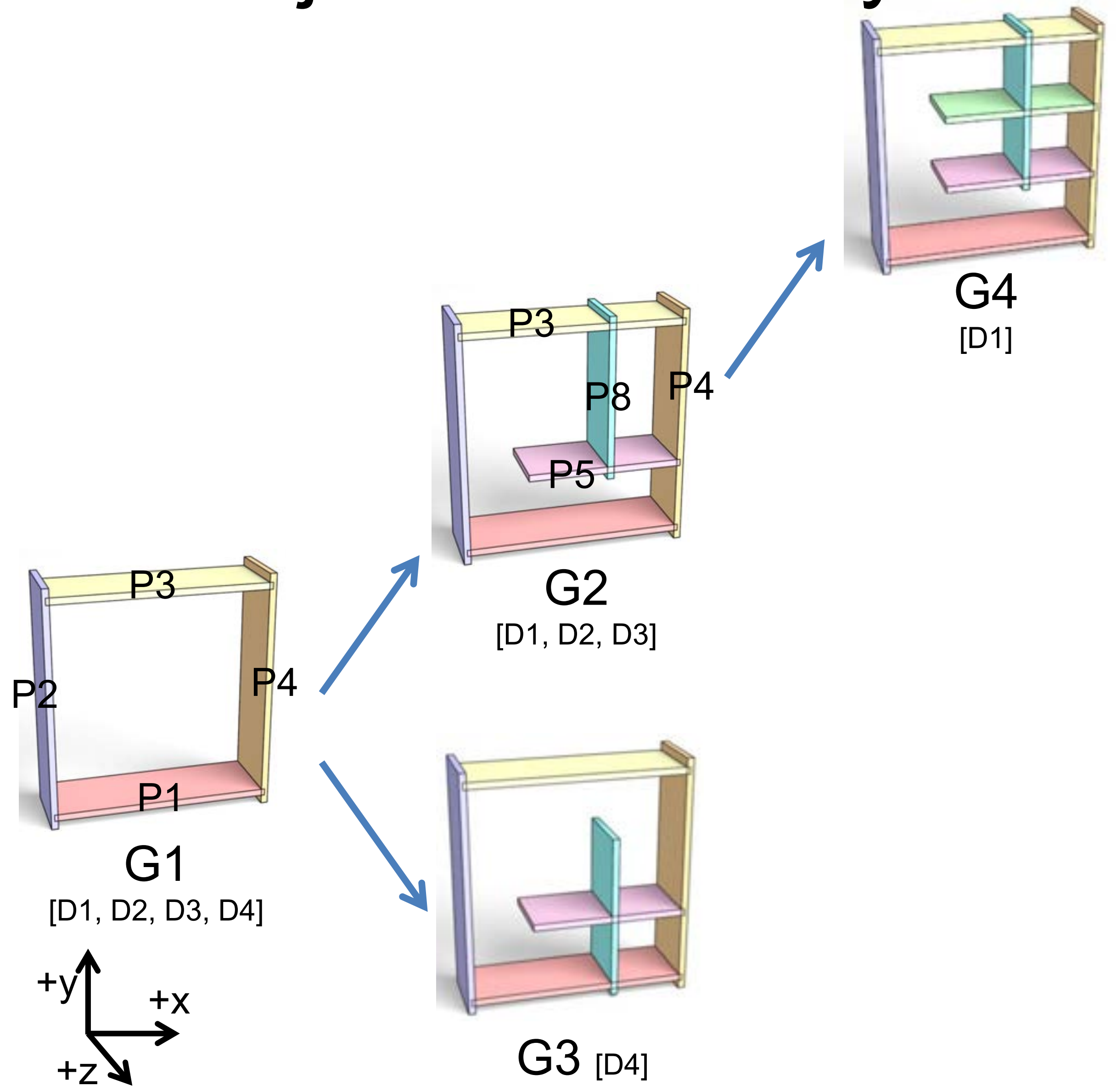
## Plan joints iteratively

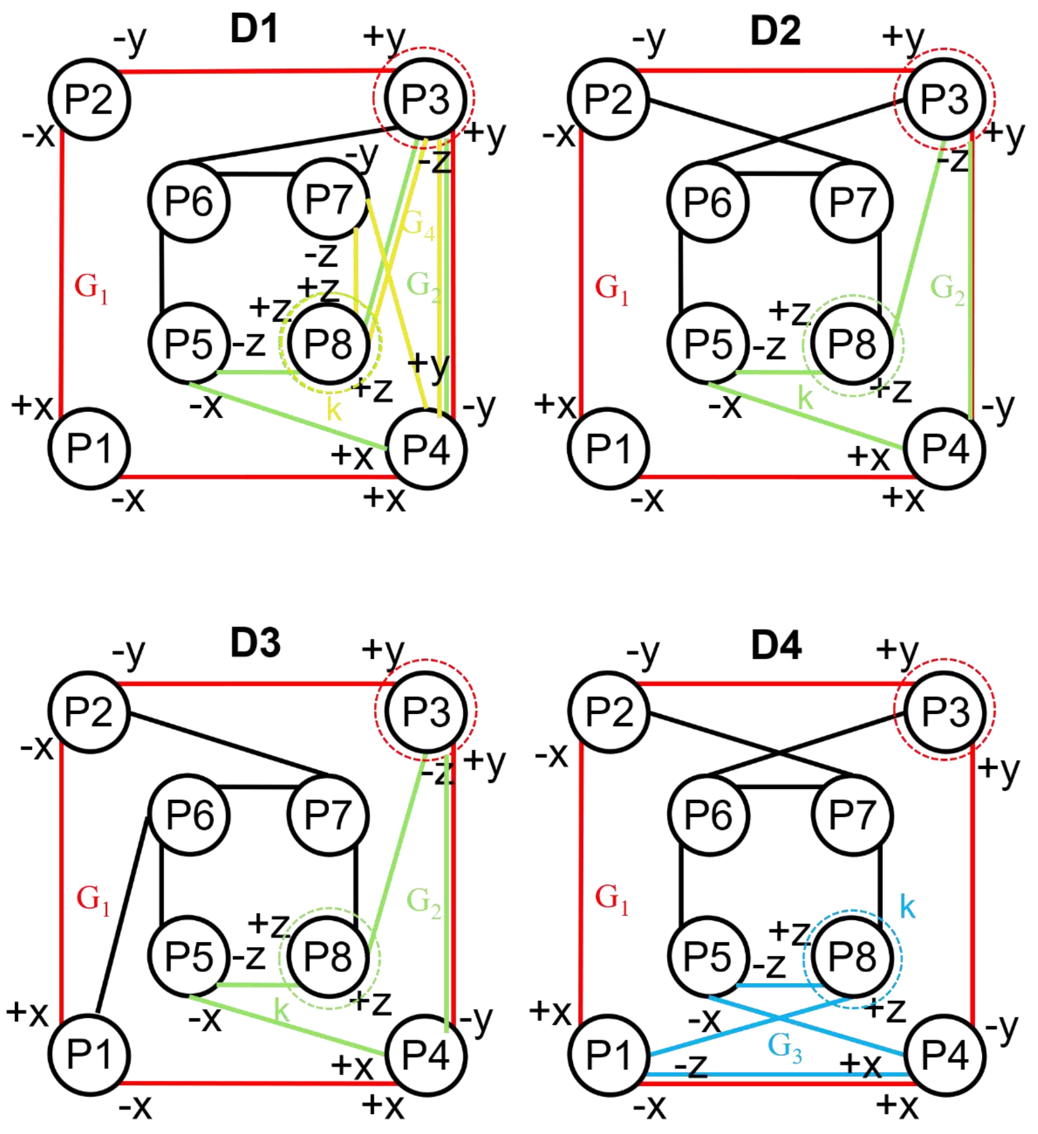


# Co-Construct Joints

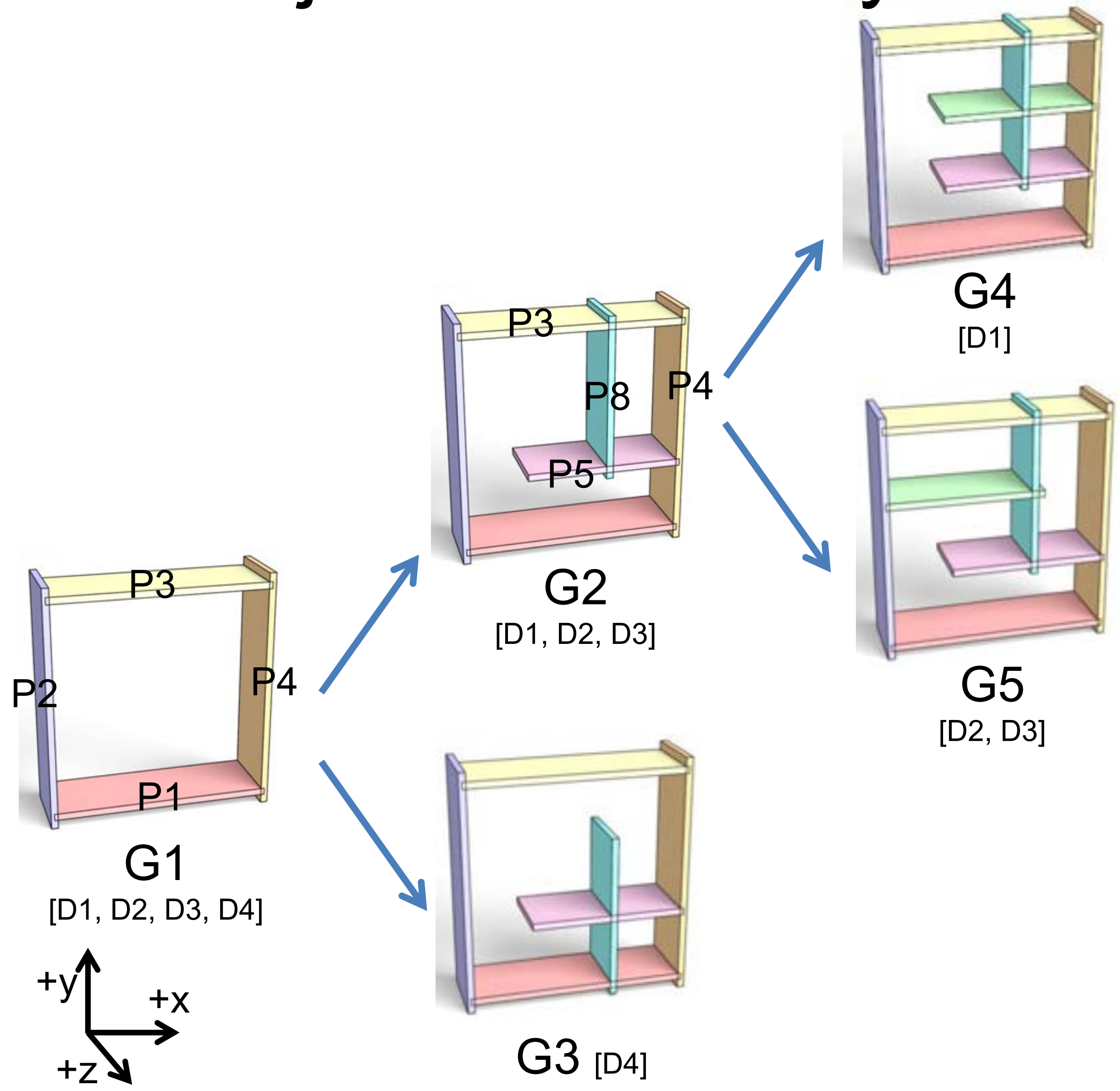
## Plan joints iteratively

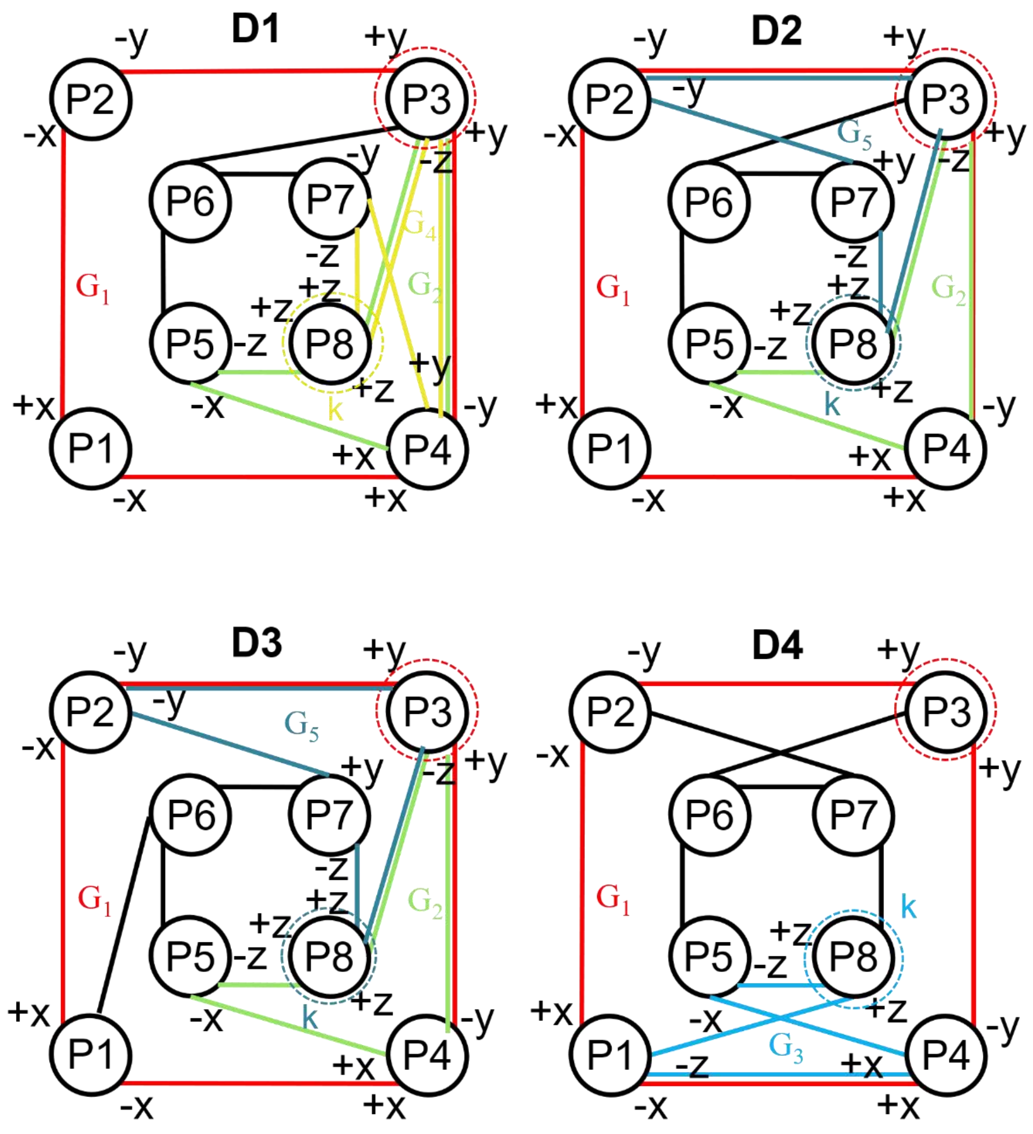


# Co-Construct Joints

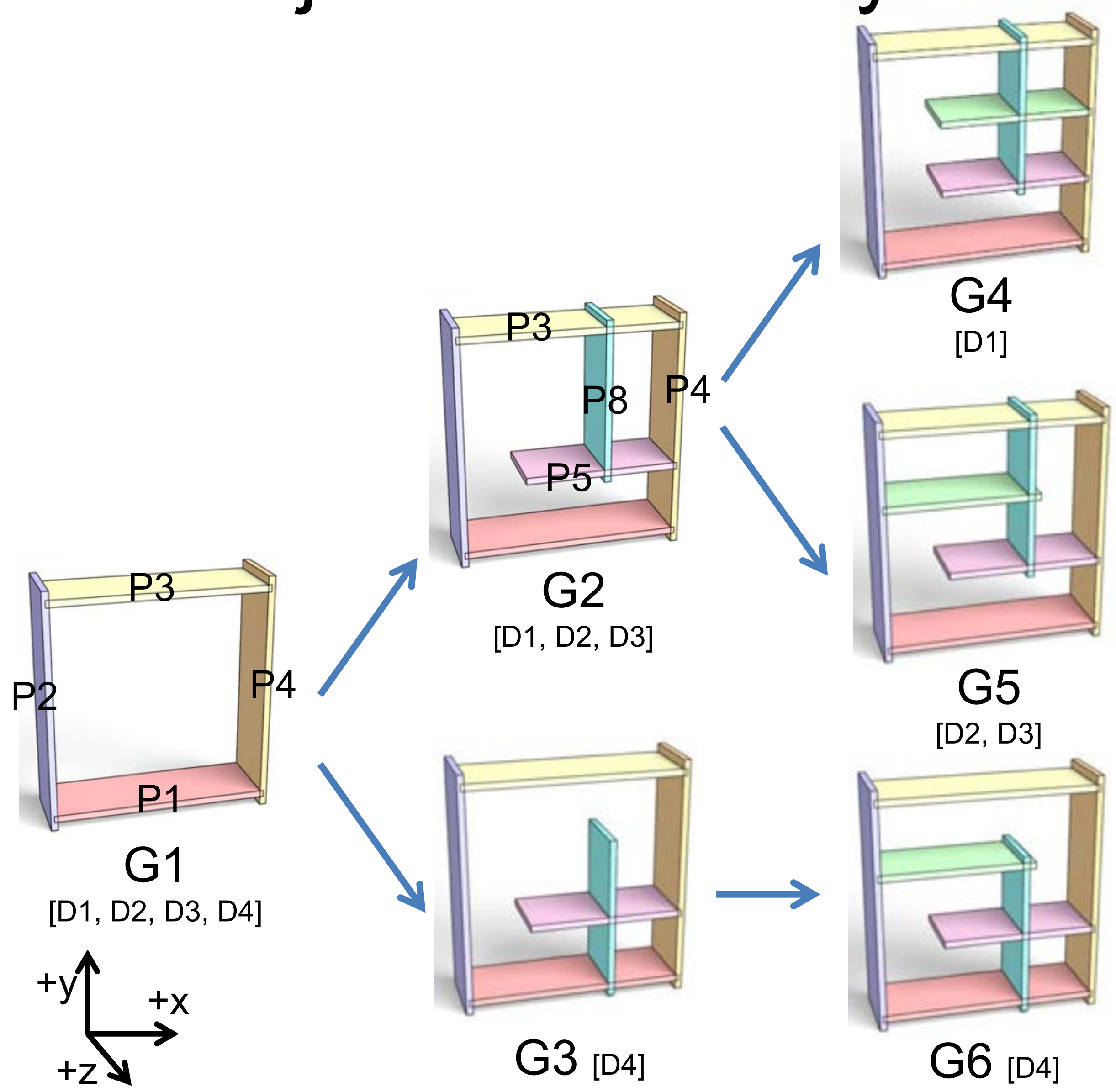
Plan joints iteratively

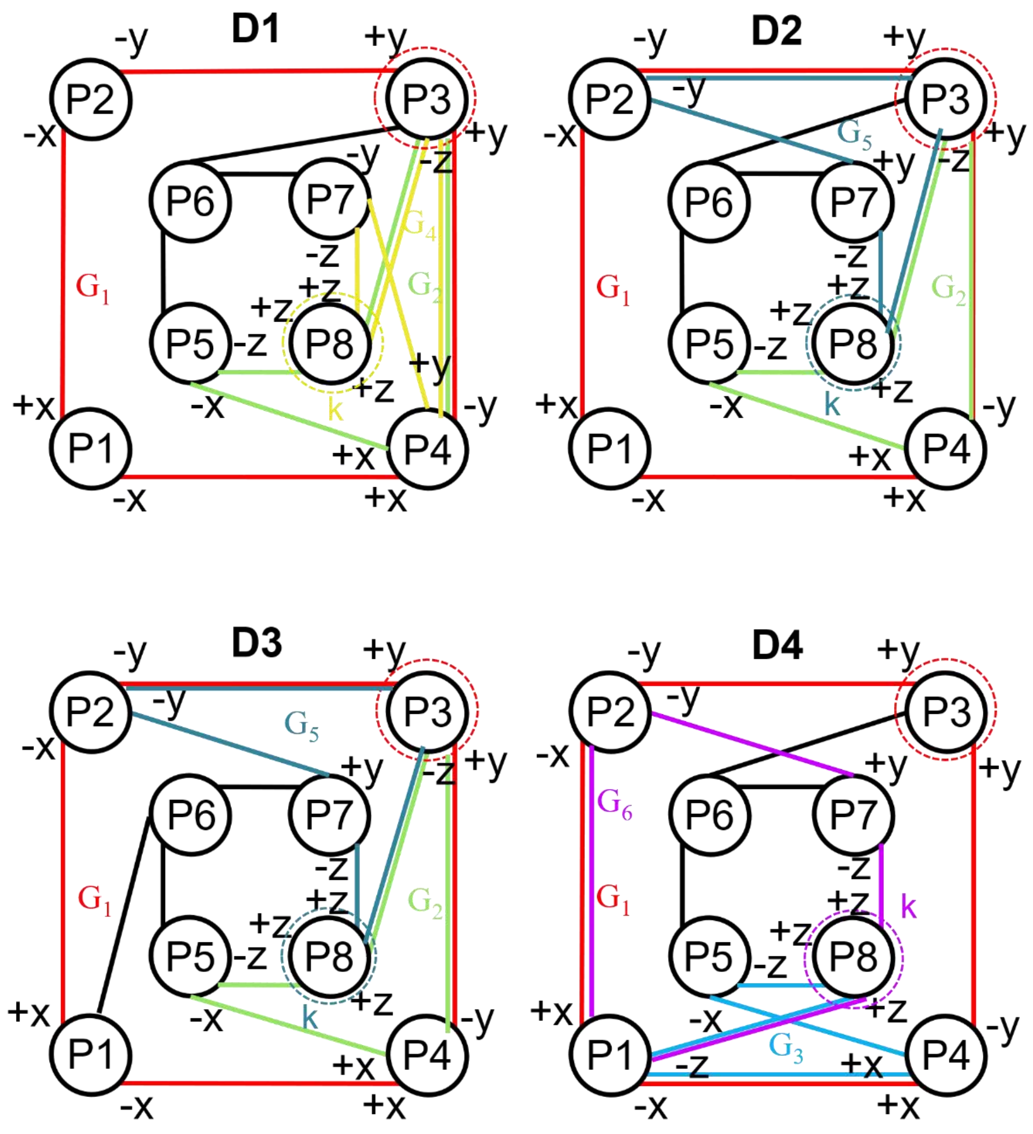


# Co-Construct Joints

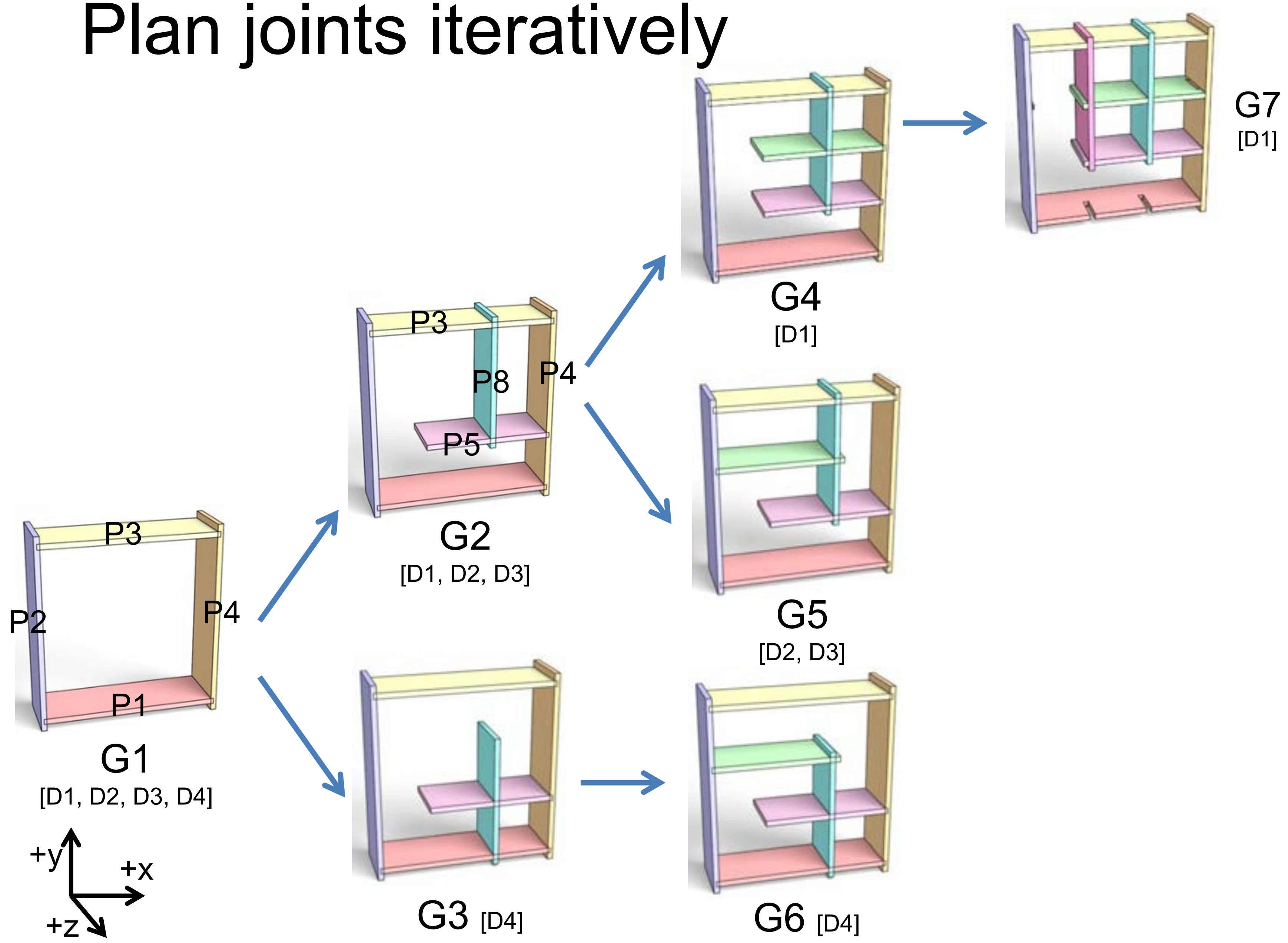
Plan joints iteratively

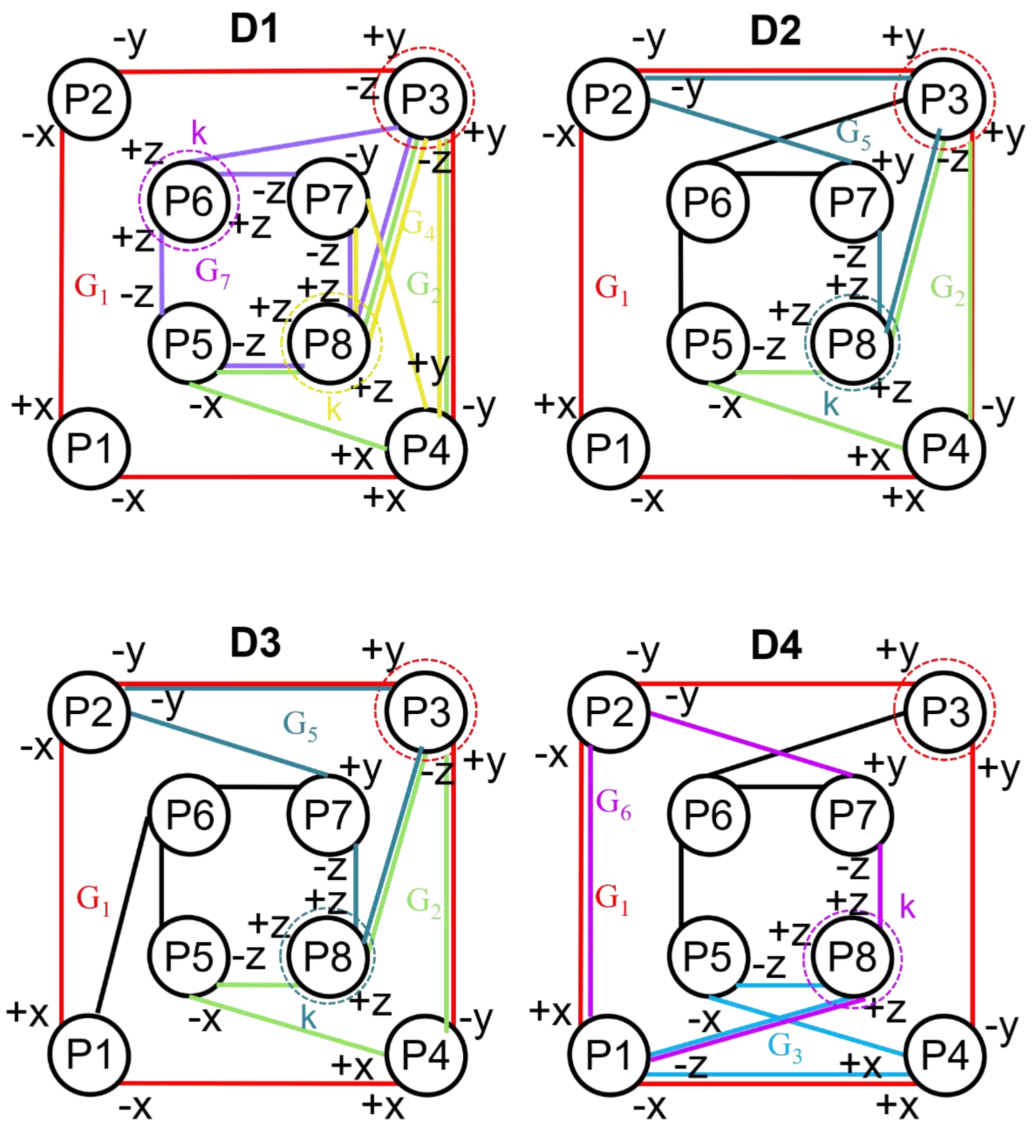


# Co-Construct Joints

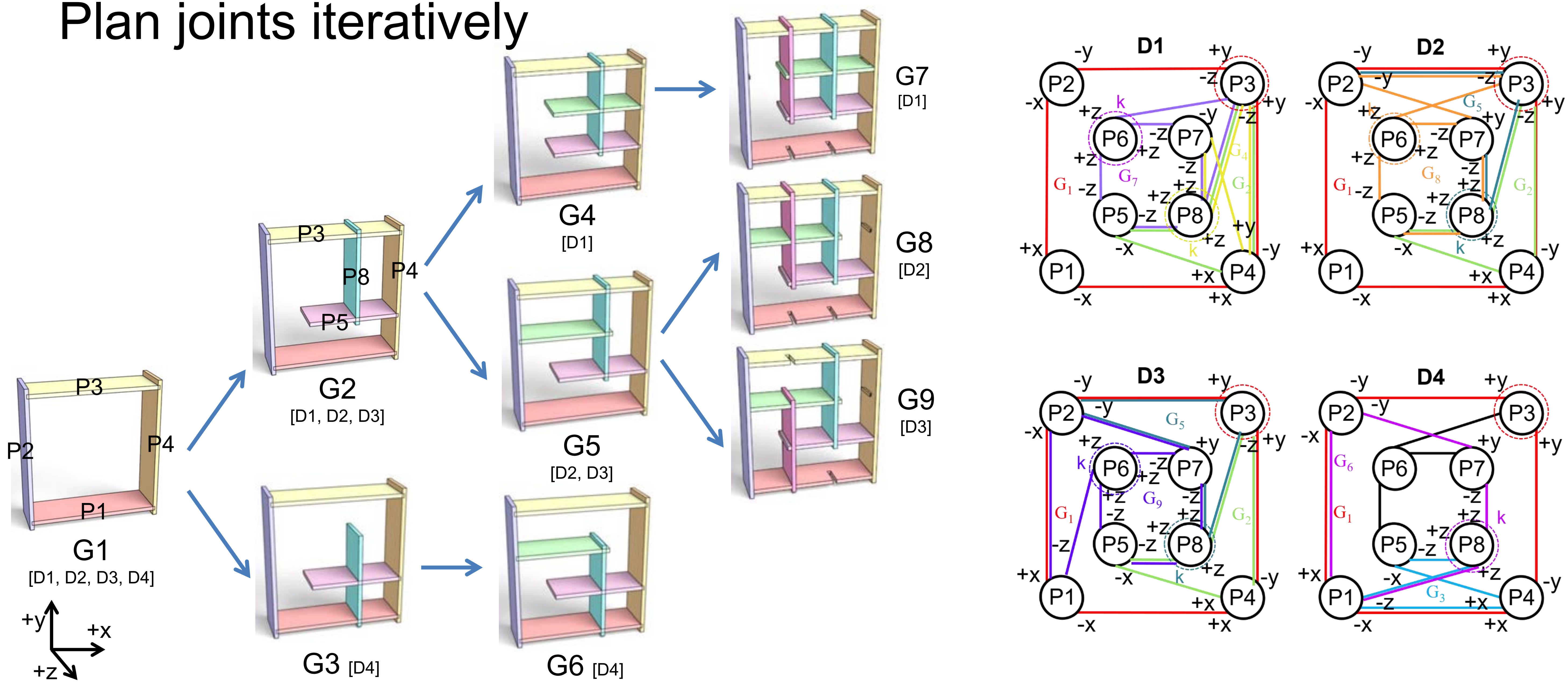
Plan joints iteratively



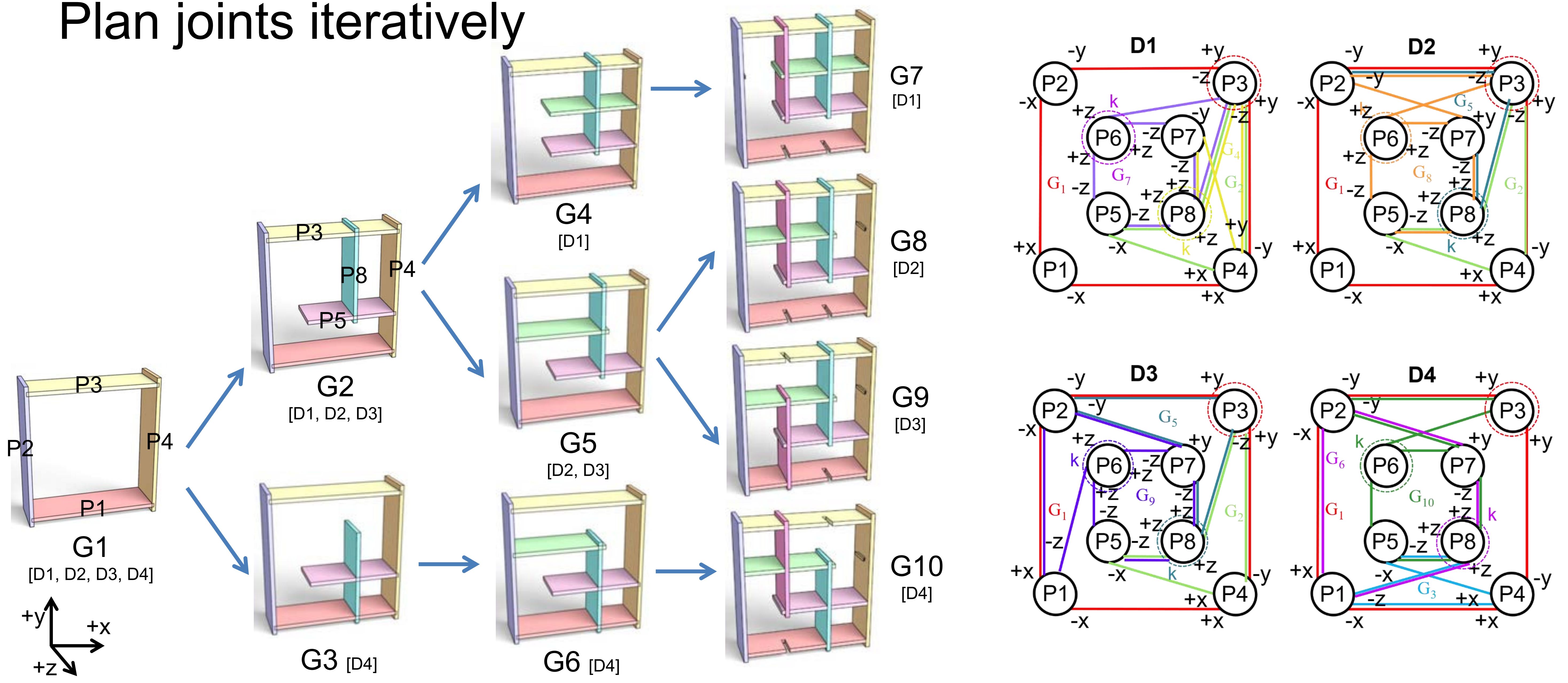
# Co-Construct Joints

Plan joints iteratively



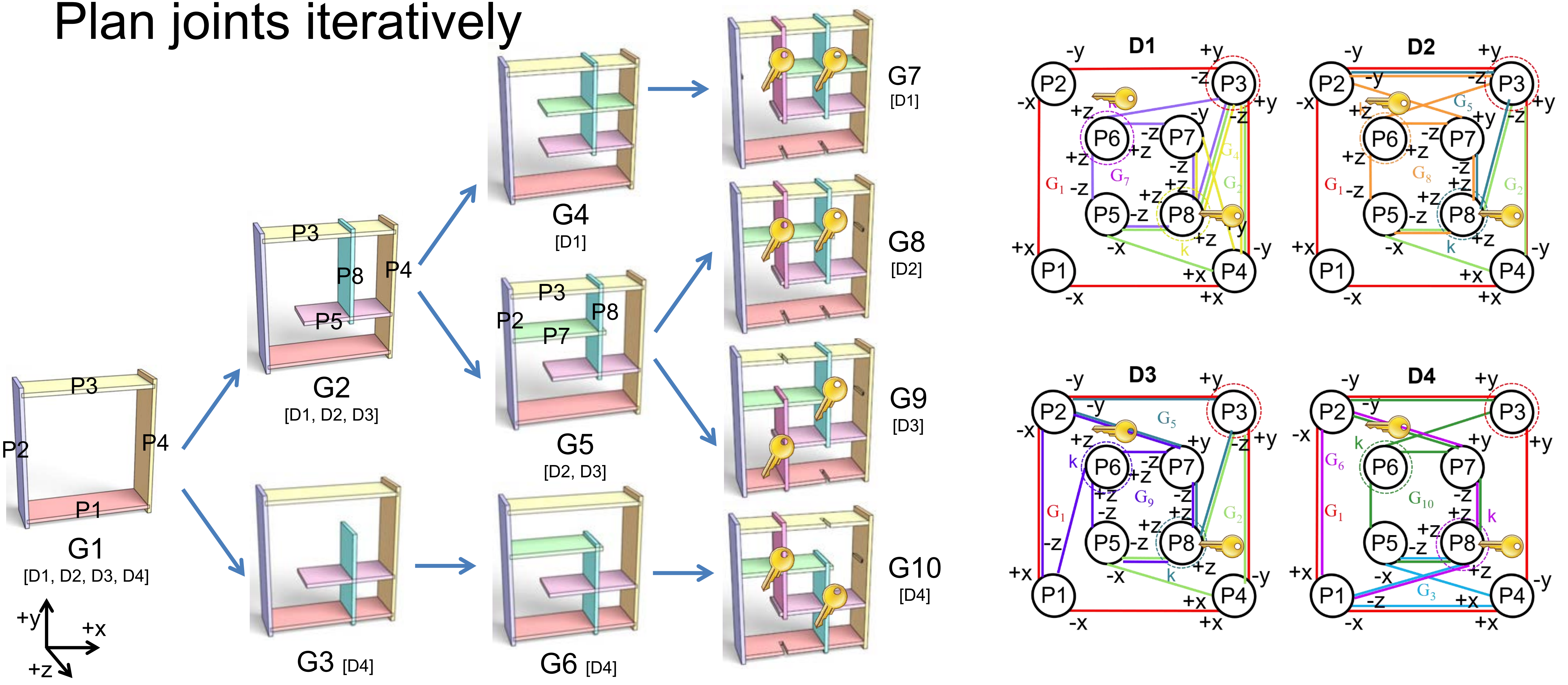
# Co-Construct Joints

## Plan joints iteratively



# Co-Construct Joints

## Plan joints iteratively



# Our Results

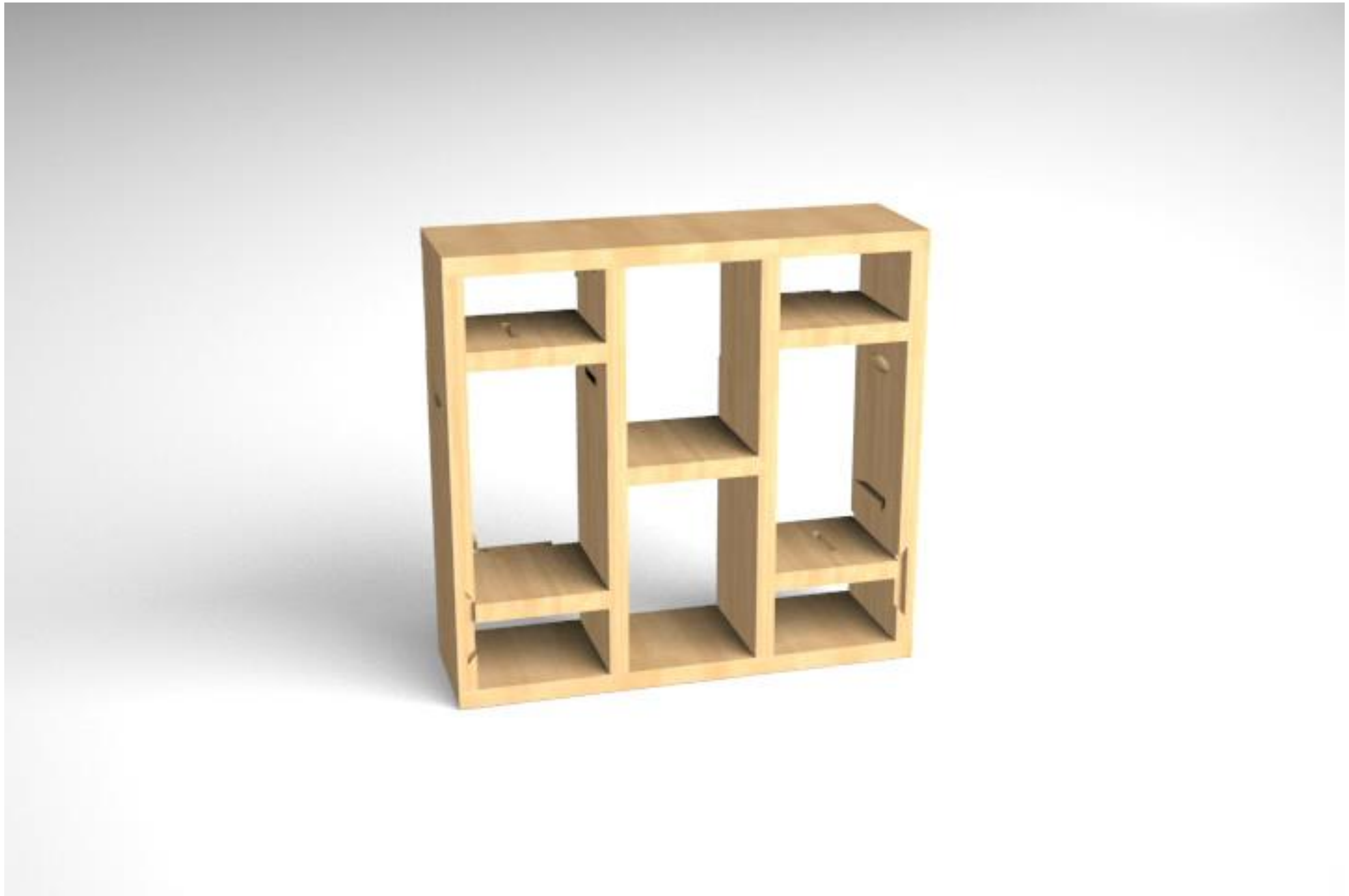
- Animation sequences showing assembly and re-assembly
- Physical fabrication: woodworking, laser cutting & 3D printing
- Extensible and hierarchical assemblies
- Baseline comparisons

# Chair - Step Ladder

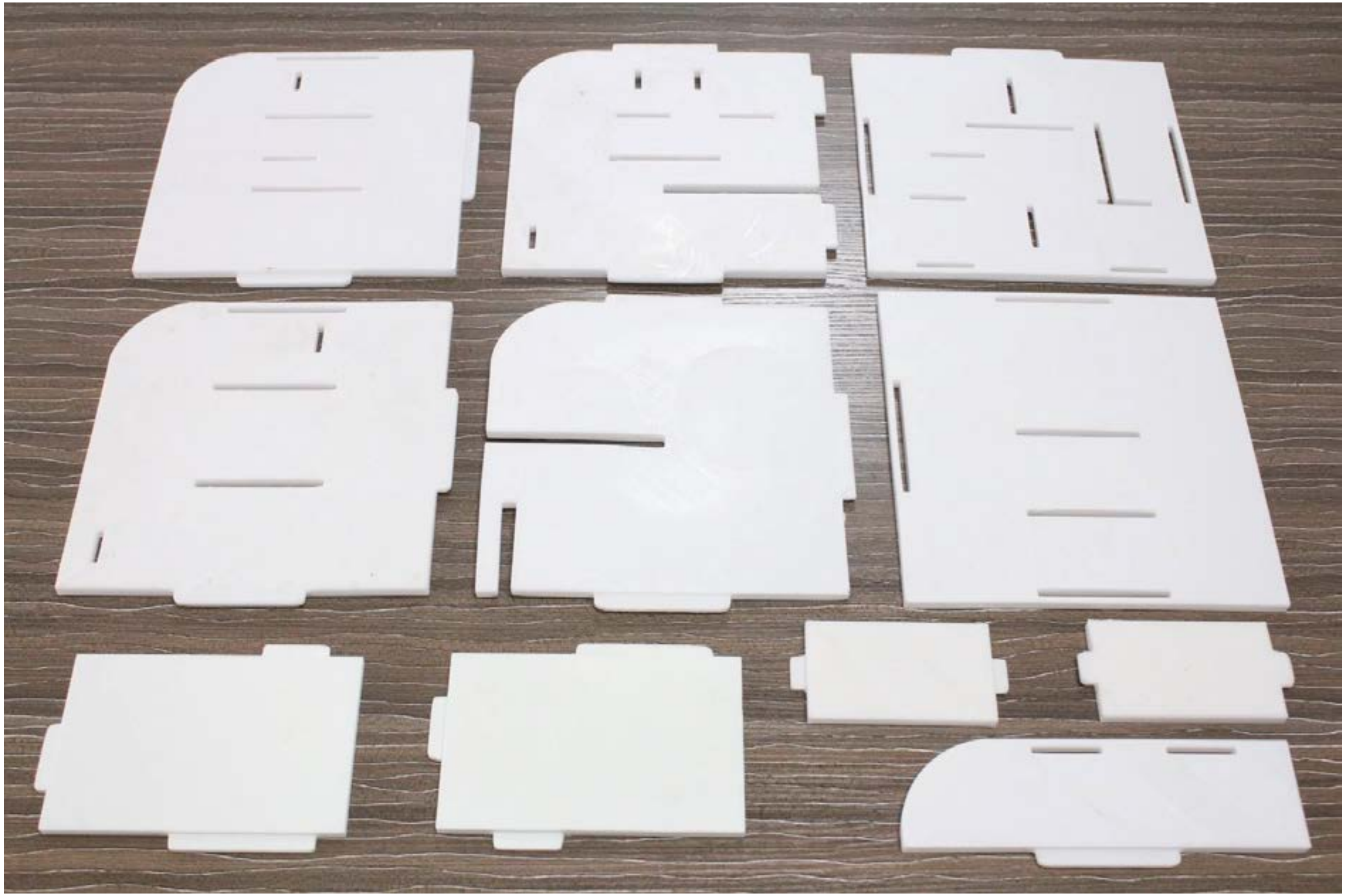
# Bookshelves

# Shoe Rack - Box

# Bookshelf - Table+Chairs



# Office Boxes



# Bookshelves

# Ladder - Truck - Stool

# Results

# Results: Statistics

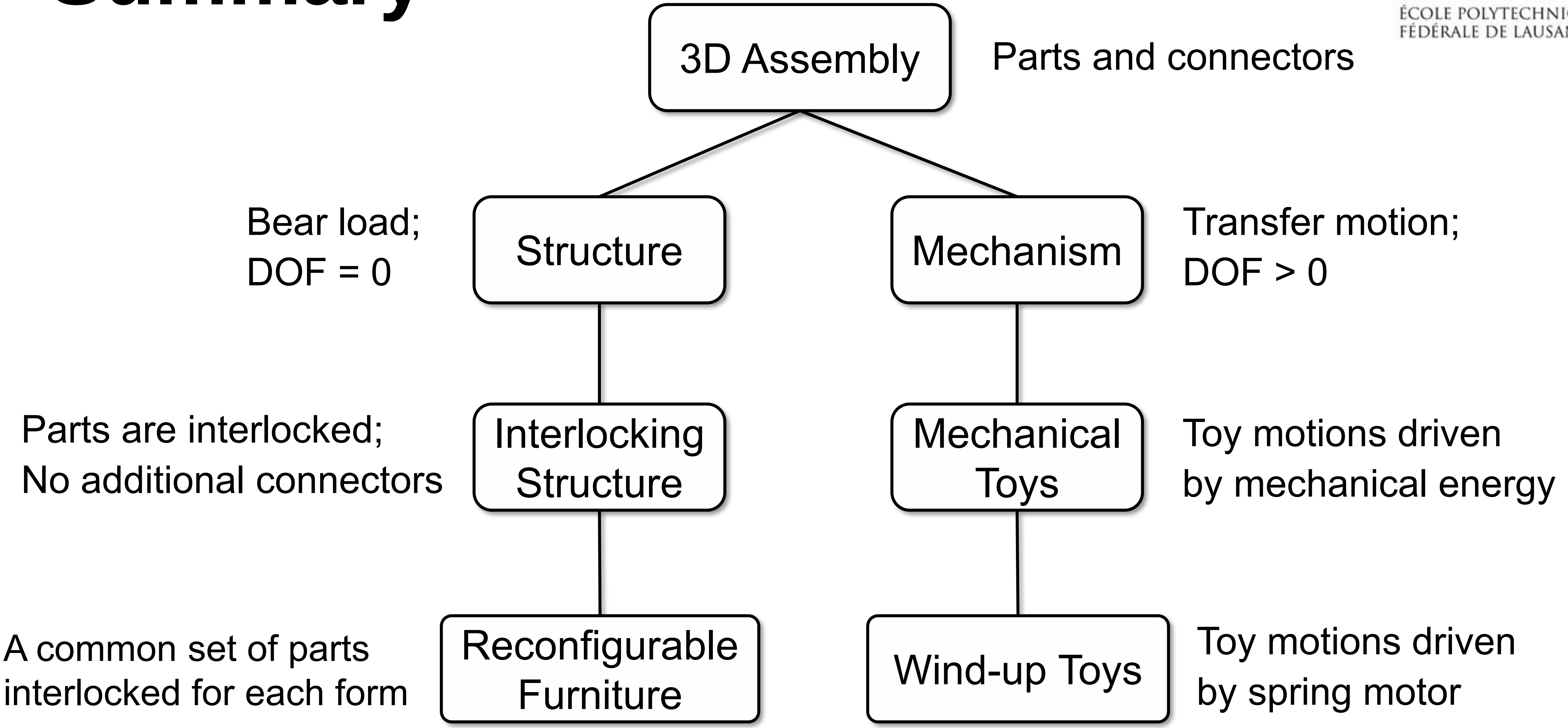


| Forms (designs) | #P | #J | #K | #C | #H | T-co-D. (sec.) | T-co-C. (sec.) |
|---|---|---|---|---|---|---|---|
| Ladder | 9 | 14 | 2 | 11 | 50 | 24.8 | 35.6 |
| Stool | 11 | 16 | 3 | | | | |
| Hand Truck | 10 | 14 | 2 | | | | |
| Step Ladder | 9 | 14 | 2 | 9 | 46 | 0.4 | 0.7 |
| Chair | 8 | 17 | 2 | | | | |
| Bookshelf 1 | 8 | 12 | 2 | 8 | 27 | 2.7 | 1.1 |
| Bookshelf 2 | 8 | 12 | 2 | | | | |
| Bookshelf 3 | 8 | 12 | 2 | | | | |
| Bookshelf 4 | 8 | 12 | 2 | | | | |
| Bed | 8 | 13 | 1 | 11 | 44 | 0.6 | 0.5 |
| Cot | 5 | 8 | 1 | | | | |
| Desk | 6 | 8 | 1 | | | | |
| Bookshelf | 15 | 22 | 1 | 16 | 70 | 275.0 | 493.8 |
| Table | 6 | 9 | 1 | | | | |
| Chair | 5 | 8 | 1 | | | | |
| Chair | 5 | 8 | 1 | | | | |
| Office Box 1 | 9 | 19 | 1 | 11 | 64 | 38.0 | 272.2 |
| Office Box 2 | 10 | 21 | 1 | | | | |
| Office Box 3 | 10 | 20 | 1 | | | | |
| Shoe Rack | 9 | 16 | 2 | 12 | 60 | 0.5 | 2.3 |
| Laundry Box | 9 | 16 | 2 | | | | |

# Limitations

- Co-decomposition process may fail if the given designs have very different dimensions or very different substructures
- Interlocking model requires the user to put in extra part(s) in case there are insufficient cyclic substructures
- Co-decomposition and co-construction processes are performed sequentially

# Summary


3D Assembly
Parts and connectors
Bear load;
DOF = 0
Structure
Mechanism
Transfer motion;
DOF > 0
Parts are interlocked;
No additional connectors
Interlocking Structure
Mechanical Toys
Toy motions driven by mechanical energy
A common set of parts interlocked for each form
Reconfigurable Furniture
Wind-up Toys
Toy motions driven by spring motor

# Summary

More information can be found at

https://songpenghit.github.io

# Acknowledgments


**Supervisors**

Chi-Wing Fu
Ligang Liu
Daniel Cohen-or
Niloy J. Mitra
Jianmin Zheng
….

**Friends**

Bailin Deng
Xiaoqi Yan
Xiaopei Liu
….

**Students**

Ziqi Wang
Zhichao Dong
Wei Li
Xiaofei Wang
Xiao Tang
Hongfei Xu
Zhongqi Fu
….

# Acknowledgments

谢谢各位讲者、老师、同学一直以来对GAMES的支持！

祝大家在新的一年身体健康、学业进步、万事如意！

2018年再见 ☺