Embodied Vision

Physical Modeling: Rigid-body Motion

Tsung-Wei Ke

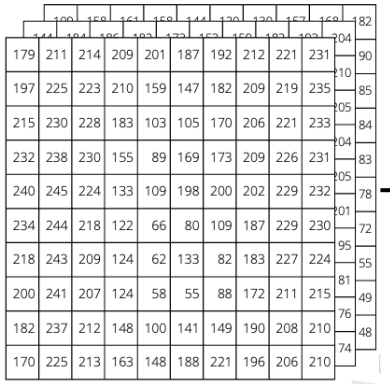
Spring 2026

Disclaimer

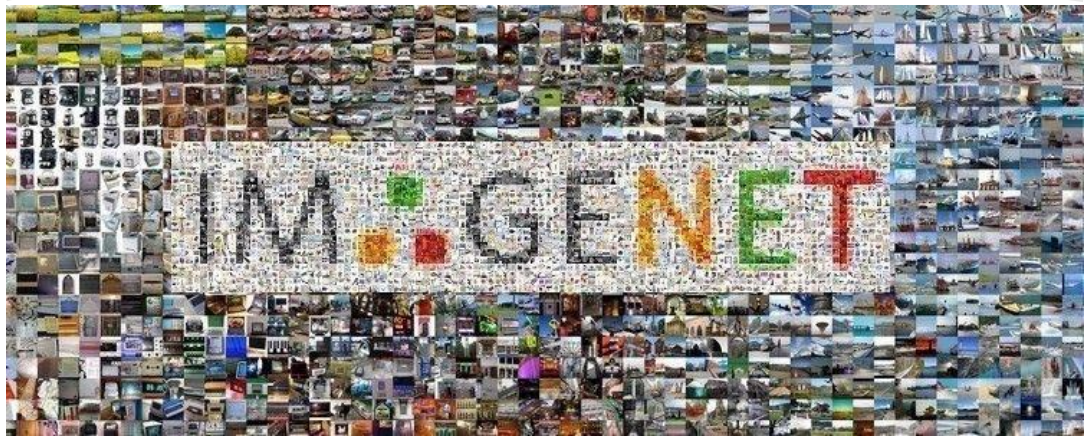
- This lecture borrows contents heavily from
  - Physics-based Animation by David I.W Levin at University of Toronto
  - Physics-based Animation of Solids and Fluids by Minchen Li at CMU

What We've Discussed Still Follows the Data-Driven Paradigm

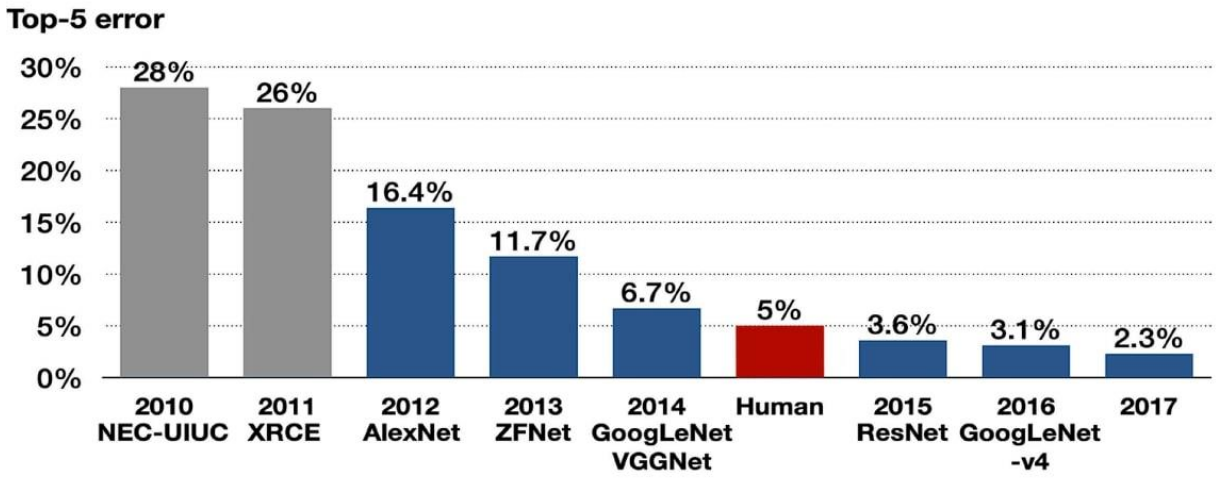
- We try to follow the same successful story: all you need is to collect more data and labels



CNN + classifier → Labels



1,000,000 images with 1,000,000 labels



Can We Solve Physics with the Data-Driven Paradigm?

- How to annotate the water wave motion?

Water wave packets. Jeschke et al.

Can We Solve Physics with the Data-Driven Paradigm?

- How to annotate the elastic-object or fluid motion?

Affine Particle In Cell. Jiang et al.
https://www.youtube.com/watch?v=rPPW_1M8nRA

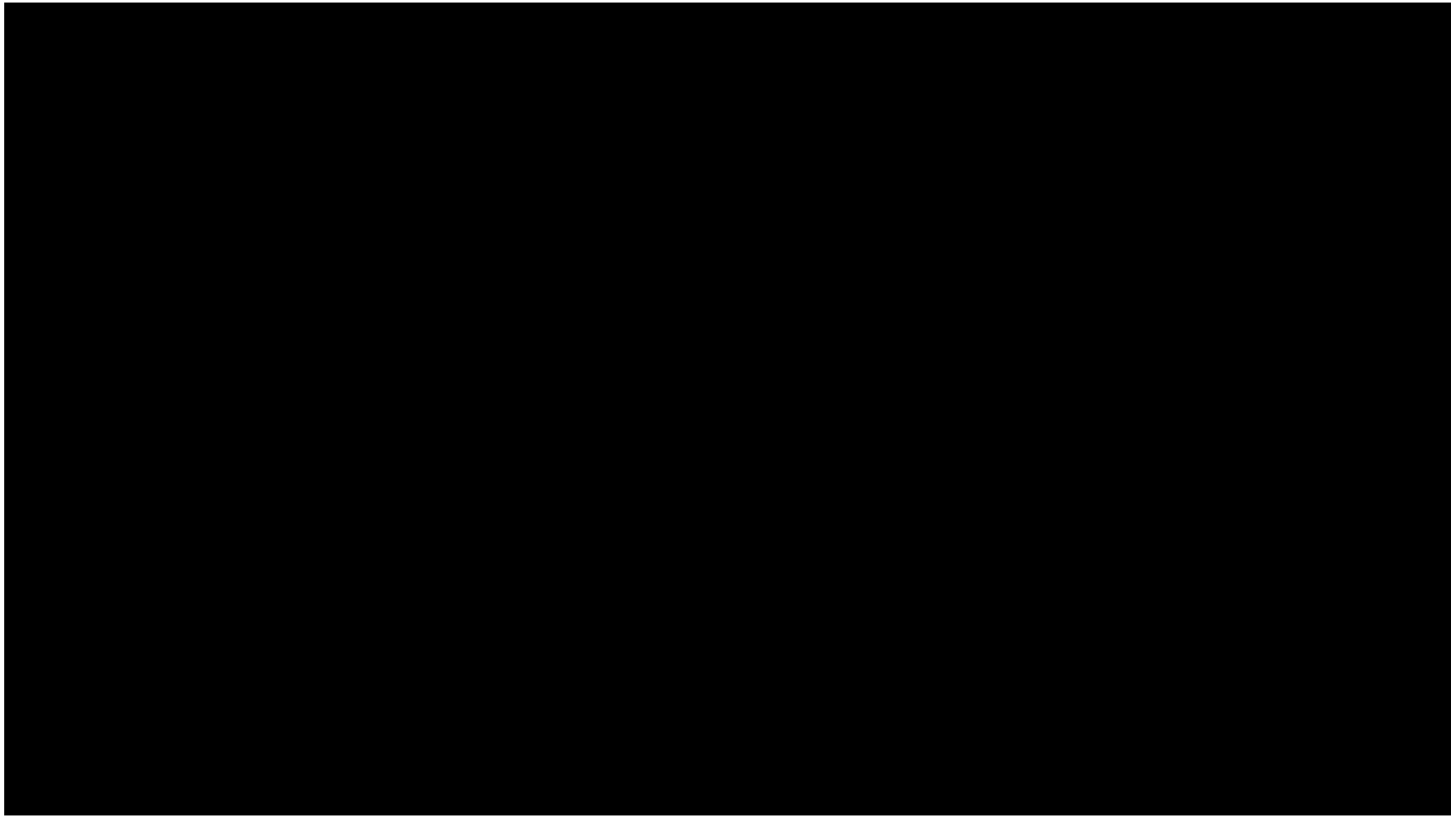
Can We Solve Physics with the Data-Driven Paradigm?

- How to annotate the motion of snow?

Disney's Frozen - A Material Point Method For Snow Simulation
https://www.youtube.com/watch?v=9H1gRQ6S7gg

Can We Solve Physics with the Data-Driven Paradigm?

- How to annotate and model the motion with friction



https://youtu.be/o94Pm-Cty3M?si=Oxn1Q68Edou4XLrn

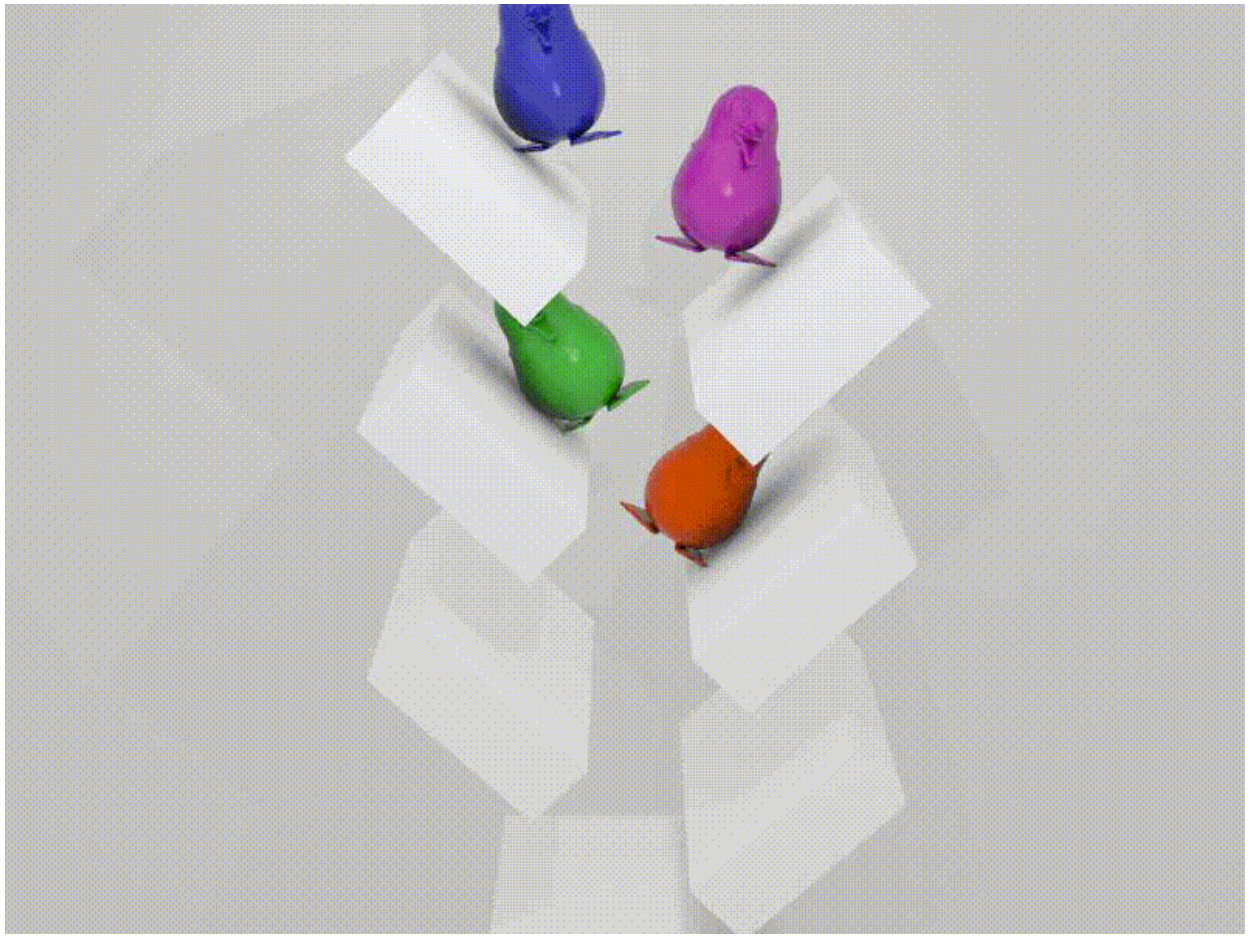
Can We Solve Physics with the Data-Driven Paradigm?

- How to model the rigid-body motion with collision

https://youtu.be/o94Pm-Cty3M?si=Oxn1Q68Edou4XLrn

Can We Solve Physics with the Data-Driven Paradigm?

- How to model the elastic-object motion with collision and friction



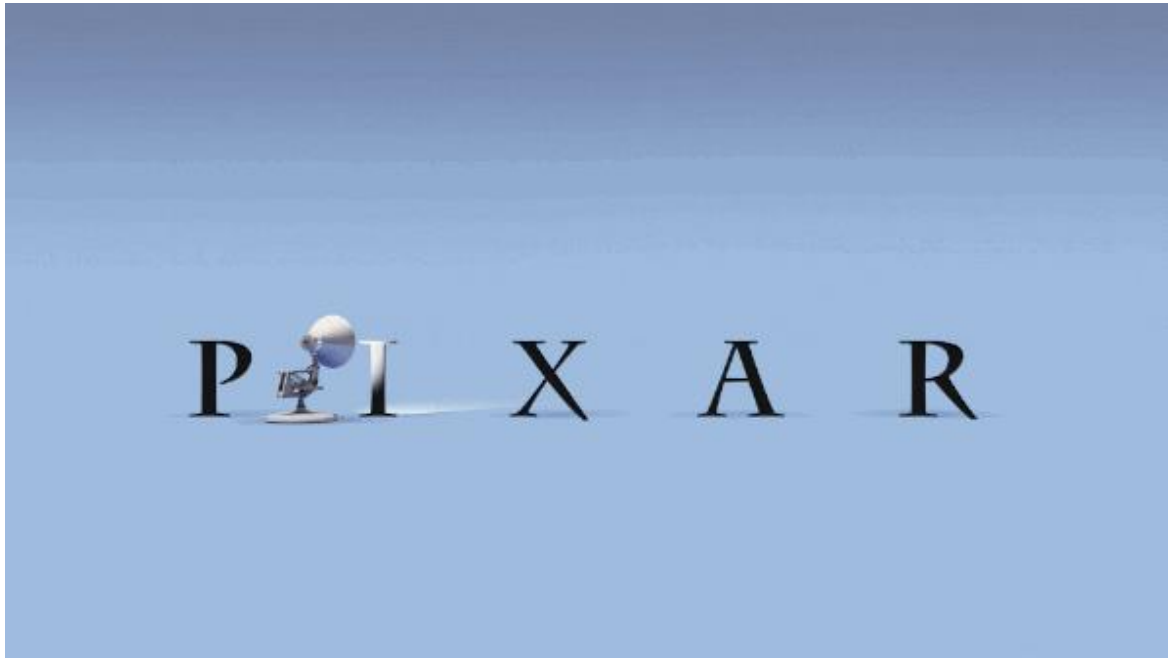
Video source: D. Levin

Can We Solve Physics with the Data-Driven Paradigm?

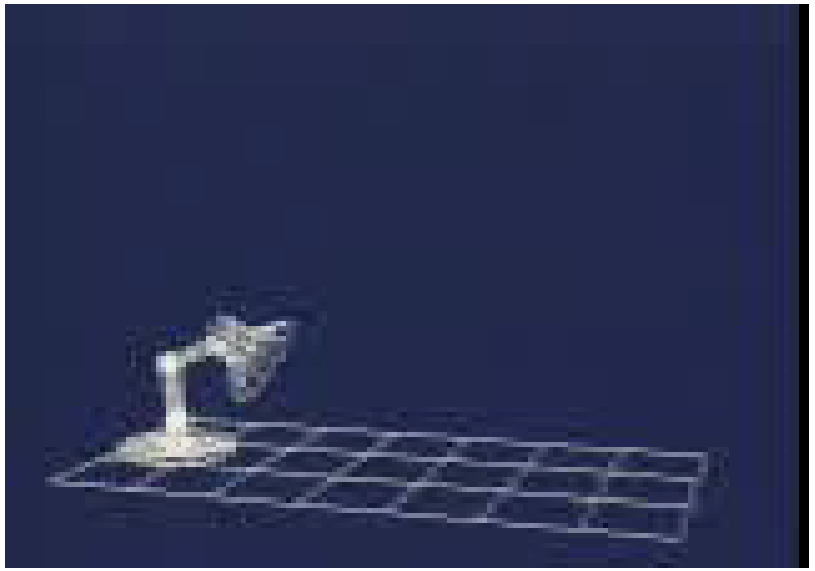
- Supervised learning methods need:
  - ➢ Data
  - ➢ Labels←It's almost impossible to label particle motion of monochromatic objects
- Unsupervised learning methods need only data, such as video generative models. However, the dynamics of generated videos are often physically incorrect.

Computer Graphics Has Solutions: Physical-Based Simulation

- Idea: Simulate motions with physic laws



Video source: Pixar



Spacetime Constraints. Witkins and Kass.

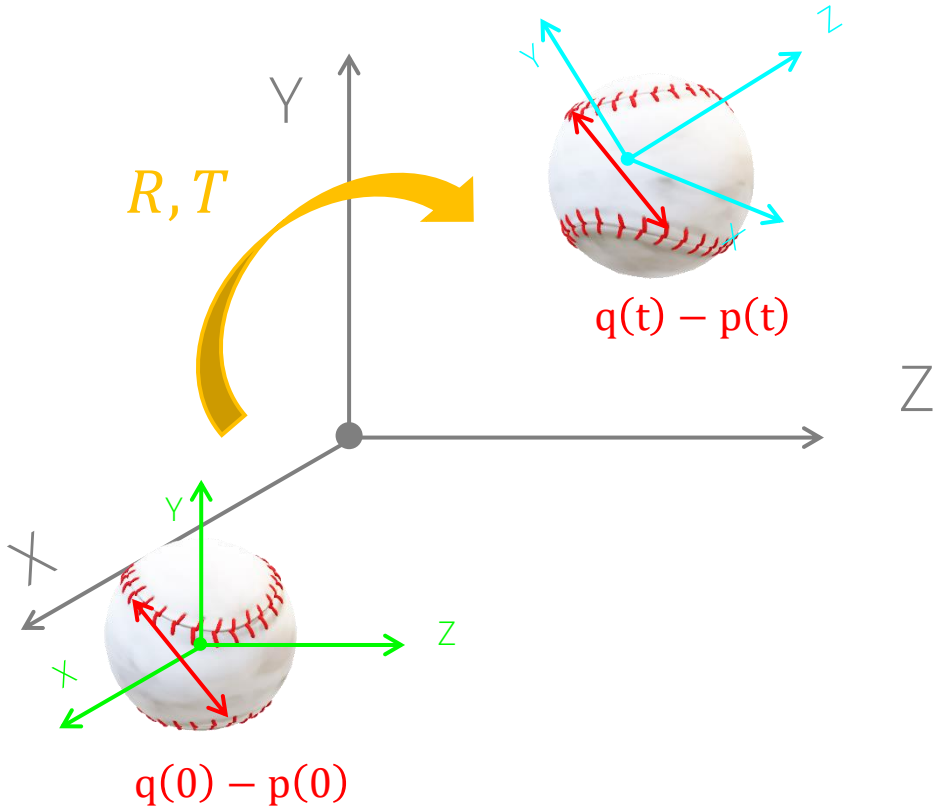
Content

- Rigid-body Modeling
  - ➢Physical law of rigid bodies
  - ➢Formulation of articulated objects
- Dynamic Modeling of Rigid Bodies
  - ➢Video generation of rigid-body motions
  - ➢Generation of 3D articulated objects

Content

- Rigid-body Modeling
  - ➢Physical law of rigid bodies
  - ➢Formulation of articulated objects
- Dynamic Modeling of Rigid Bodies
  - ➢Video generation of rigid-body motions
  - ➢Generation of 3D articulated objects

Rigid-Body Motions



- Rigid body: A rigid body is a collection of particles s.t. the distance between any 2 particles are fixed:

$$\|\mathbf{q}(\mathbf{t}) - \mathbf{p}(\mathbf{t})\| - \|\mathbf{q}(\mathbf{0}) - \mathbf{p}(\mathbf{0})\|$$

- Every particle motion can be described by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{T}$

Rigid-Body Physic Law

Rigid-body physic law:

$$\frac{d}{dt}\mathbf{q}(t) = \frac{d}{dt}\begin{bmatrix}\mathbf{t}(t)\\ \mathbf{R}(t)\\ \boldsymbol{\nu}(t)\\ \boldsymbol{\omega}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{v}(t)\\ \boldsymbol{\omega}(t)\times\mathbf{R}(t)\\ \frac{\mathbf{F}(t)}{M}\\ \mathbf{I}(t)^{-1}(\boldsymbol{\tau}-\boldsymbol{\omega}(t)\times\mathbf{I}(t)\boldsymbol{\omega}(t))\end{bmatrix}$$

Simulation:

$$\mathbf{q}(t) = \mathbf{q}(0) + \int_0^t \frac{d}{dt}\mathbf{q}(t)dt = \mathbf{q}(0) + \sum_{i=1}^{T} \frac{d}{dt}\mathbf{q}(t)|_{t=t_i}\Delta_t$$

We will cover in the next lecture

Kinematics of Rotation $\frac{d\boldsymbol{R}(\mathrm{t})}{d\mathrm{t}} = \omega(\mathrm{t}) \times \mathrm{R}(\mathrm{t})$

- Let $\mathbf{r_{body}}$ be a constant, fixed point on the rigid body relative to its center of mass. Its position in the fixed world frame is $\mathbf{r_{world}}(\mathrm{t}) = \mathrm{R}(\mathrm{t})\mathbf{r_{body}}$
- If the body is rotating with an angular velocity $\omega(\mathrm{t})$, the linear velocity of that point in the world frame is $\dot{\mathbf{r}}_{\mathbf{world}}(\mathrm{t}) = \omega(\mathrm{t}) \times \mathbf{r_{world}}(\mathrm{t})$
- We then have

$$\frac{d\mathbf{r_{world}}}{d\mathrm{t}} = \frac{d\mathrm{R}(\mathrm{t})\mathbf{r_{body}}}{d\mathrm{t}} = \omega(\mathrm{t}) \times \mathbf{r_{world}}(\mathrm{t}) = \omega(\mathrm{t}) \times \mathrm{R}(\mathrm{t})\mathbf{r_{body}}$$

$$\Rightarrow \dot{\mathrm{R}}(\mathrm{t})\mathbf{r_{body}} = \omega(\mathrm{t}) \times \mathrm{R}(\mathrm{t})\mathbf{r_{body}}$$

$$\Rightarrow \frac{d\boldsymbol{R}(\mathrm{t})}{d\mathrm{t}} = \omega(\mathrm{t}) \times \mathrm{R}(\mathrm{t})$$

Dynamics of Rotation $\frac{d\boldsymbol{\omega}(\mathrm{t})}{d\mathrm{t}} = \mathbf{I}(\mathrm{t})^{-1}\left(\boldsymbol{\tau} - \boldsymbol{\omega}(\mathrm{t}) \times \mathbf{I}(\mathrm{t})\boldsymbol{\omega}(\mathrm{t})\right)$

- Angular momentum in the world frame is the product of the world-frame inertia tensor $\mathbf{I}(\mathrm{t})$ and the angular velocity $\boldsymbol{\omega}(\mathrm{t})$: $\mathbf{L}(\mathrm{t}) = \mathbf{I}(\mathrm{t})\boldsymbol{\omega}(\mathrm{t})$
- The net torque $\boldsymbol{\tau}$ equals the rate of change of angular momentum

$$\boldsymbol{\tau} = \frac{d\mathbf{L}(\mathrm{t})}{d\mathrm{t}} = \dot{\mathbf{I}}(\mathrm{t})\boldsymbol{\omega}(\mathrm{t}) + \mathbf{I}(\mathrm{t})\dot{\boldsymbol{\omega}}(\mathrm{t})$$

Dynamics of Rotation $\frac{d\omega(t)}{dt} = I(t)^{-1}\left(\tau - \omega(t) \times I(t)\omega(t)\right)$

- Angular momentum in the world frame is the product of the world-frame inertia tensor $I(t)$ and the angular velocity $\omega(t)$: $L(t) = I(t)\omega(t)$
- The net torque τ equals the rate of change of angular momentum

$$\tau = \frac{dL(t)}{dt} = \dot{I}(t)\omega(t) + I(t)\dot{\omega}(t)$$

- The inertia tensor changes as the object rotates in the world frame

$$I(t) = R(t)I_{body}R(t)^T \Rightarrow \frac{dI(t)}{dt} = \frac{dR(t)}{dt} I_{body}R(t)^T + R(t)I_{body}\frac{dR(t)^T}{dt}$$

$$\Rightarrow \dot{I}(t) = \omega(t) \times R(t)I_{body}R(t)^T + R(t)I_{body}\left(\omega(t) \times R(t)\right)^T$$

$$\Rightarrow \dot{I}(t) = \omega(t) \times I(t) - R(t)I_{body}R(t)^T \times \omega(t)$$

$$\Rightarrow \dot{I}(t) = \omega(t) \times I(t) - I(t) \times \omega(t)$$

Dynamics of Rotation $\frac{d\omega(t)}{dt} = I(t)^{-1}\left(\tau - \omega(t) \times I(t)\omega(t)\right)$

- The inertia tensor changes as the object rotates in the world frame

$$\dot{I}(t) = \omega(t) \times I(t) - I(t) \times \omega(t)$$

$$\Rightarrow \dot{I}(t)\omega(t) = \omega(t) \times I(t)\omega(t) - I(t) \times \omega(t)\omega(t)$$

- Then we have

$$\tau = \dot{I}(t)\omega(t) + I(t)\dot{\omega}(t) = \omega(t) \times I(t)\omega(t) + I(t)\dot{\omega}(t)$$

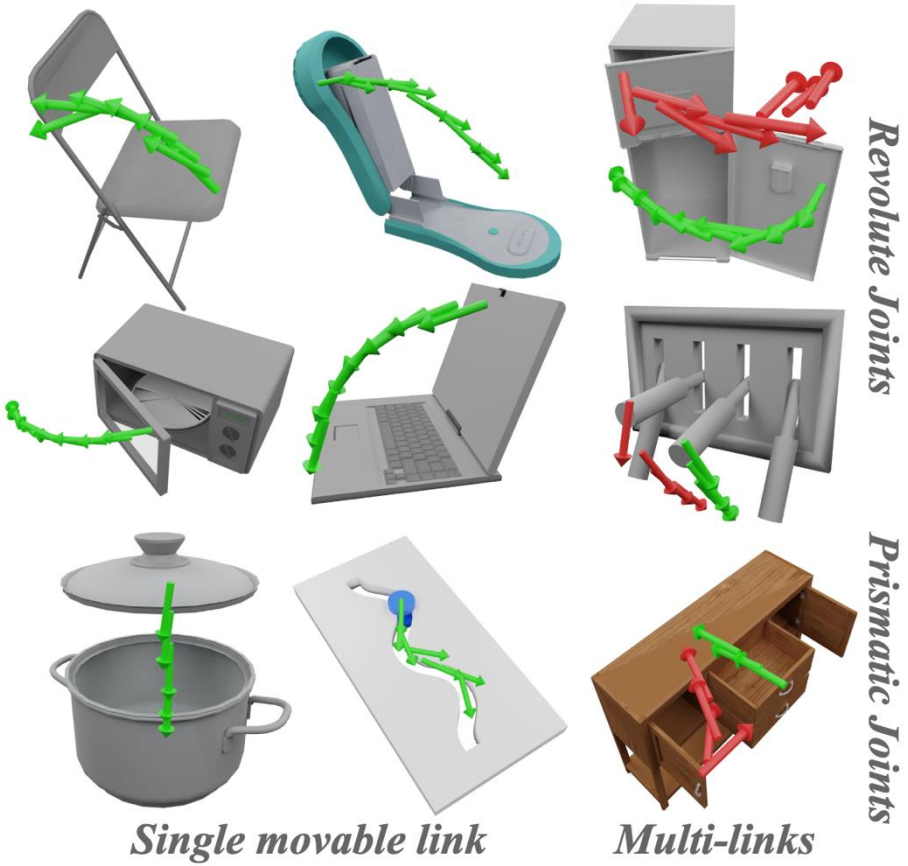
$$\Rightarrow \dot{\omega}(t) = I(t)^{-1}\left(\tau - \omega(t) \times I(t)\omega(t)\right)$$

Check “A Mathematical Introduction to Robotic Manipulation” by Richard M. Murray, Zexiang Li and S. Shankar Sastry for more details

Content

- Rigid-body Modeling
  - ➢Physical law of rigid bodies
  - ➢Formulation of articulated objects
- Dynamic Modeling of Rigid Bodies
  - ➢Video generation of rigid-body motions
  - ➢Generation of 3D articulated objects

What are Articulated Objects

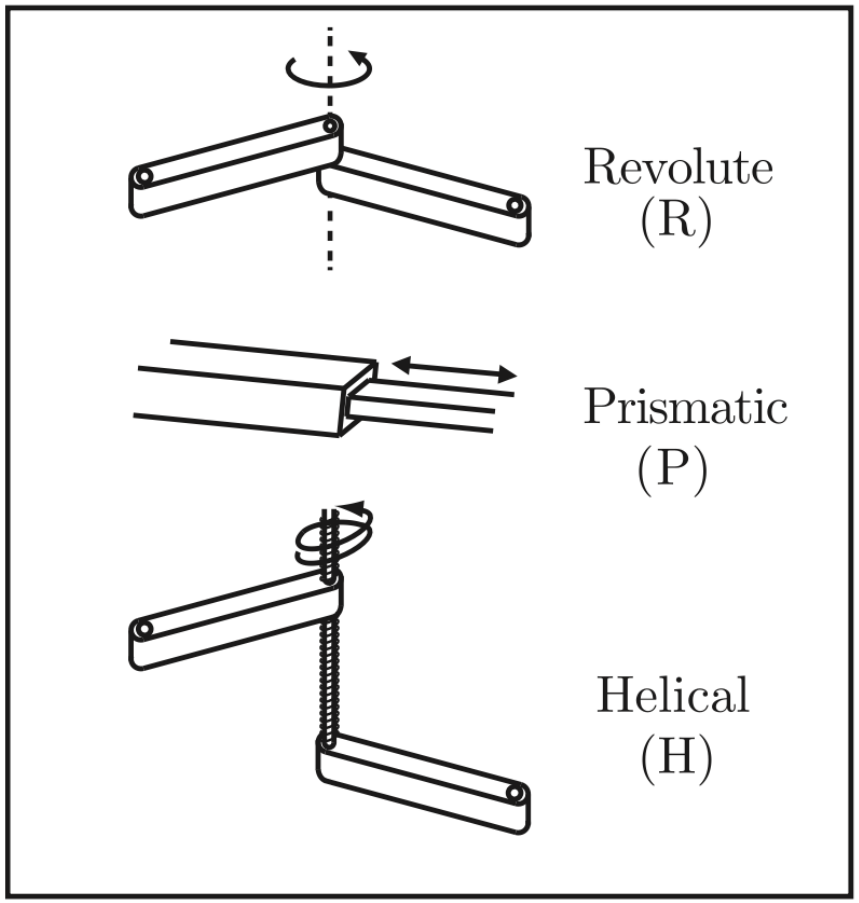


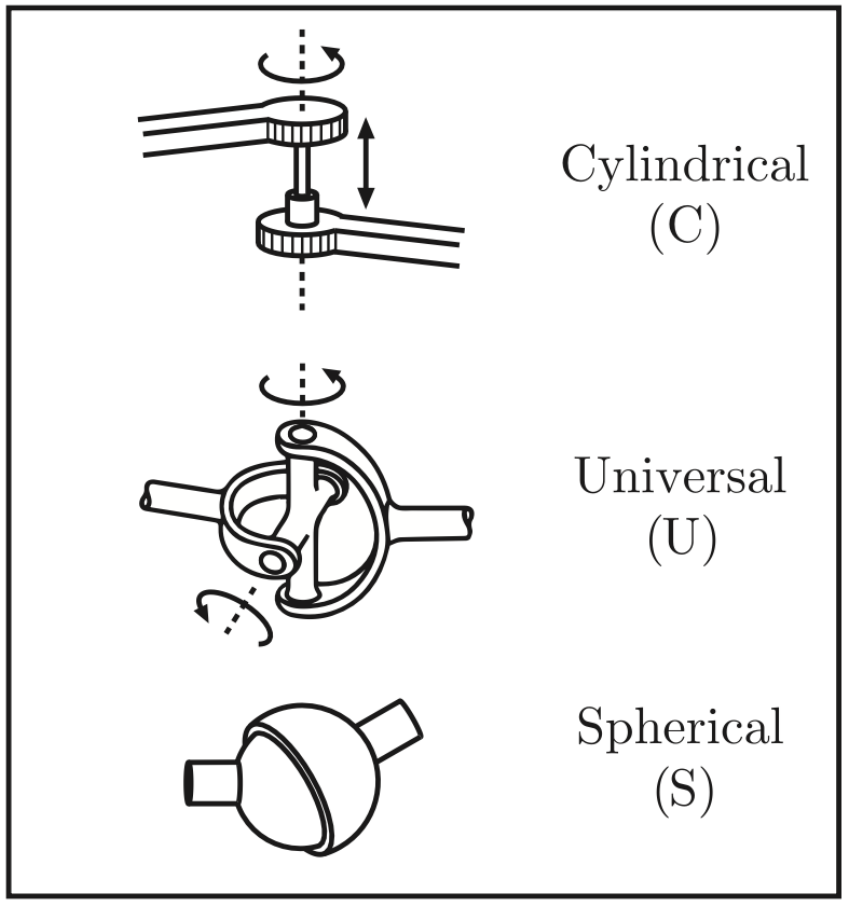
- Articulated objects: objects composed of multiple rigid bodies which move relative to one another
- Their relative motions are characterized by the “joint”
- We can describe the motion of one part by chaining all relative motions of its root parts

UMPNet: Universal Manipulation Policy Network for Articulated Objects. Xu et al

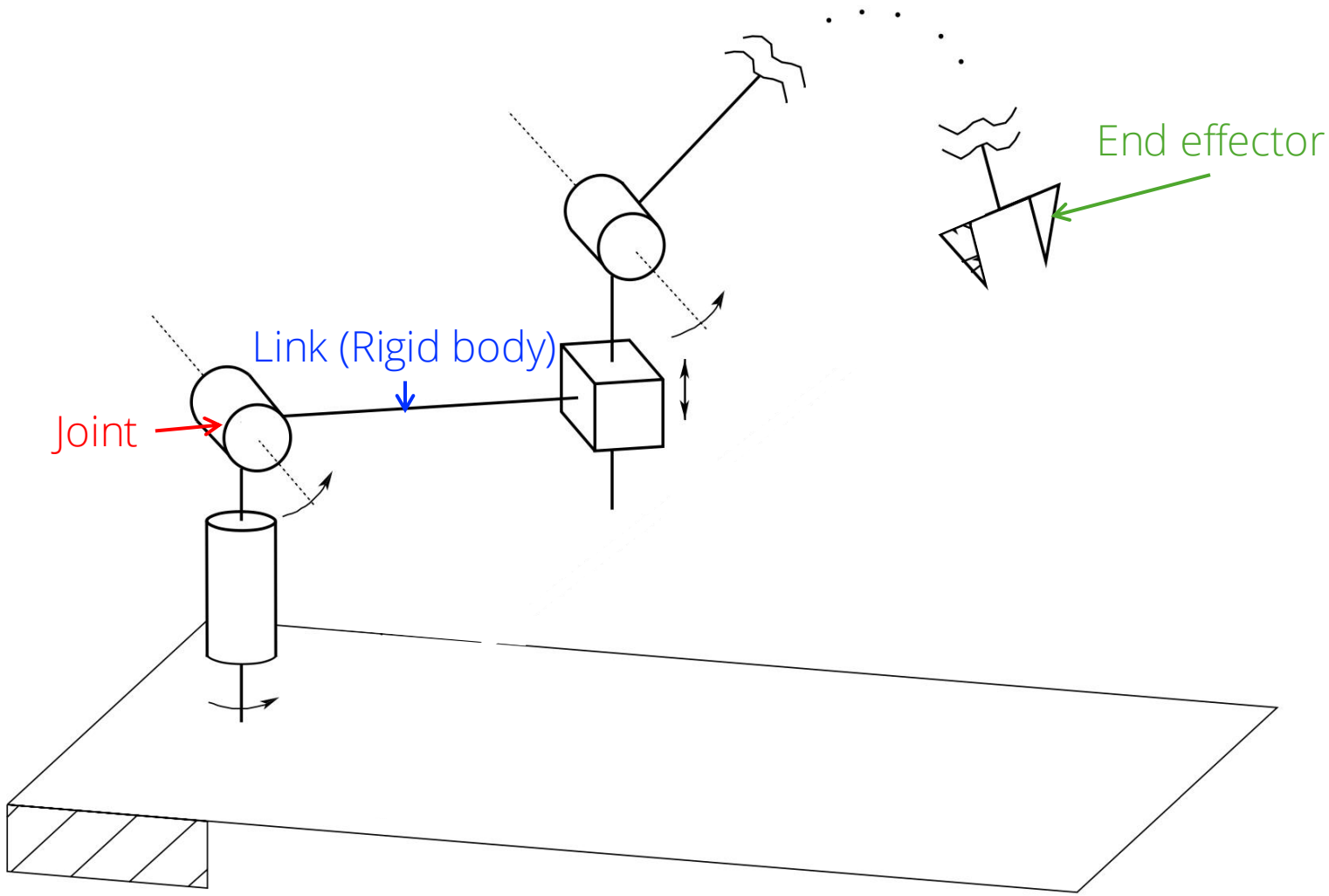
Typical Joints

Relative motions of parts can be described by the configuration of each joint (e.g. the angle of the joint)



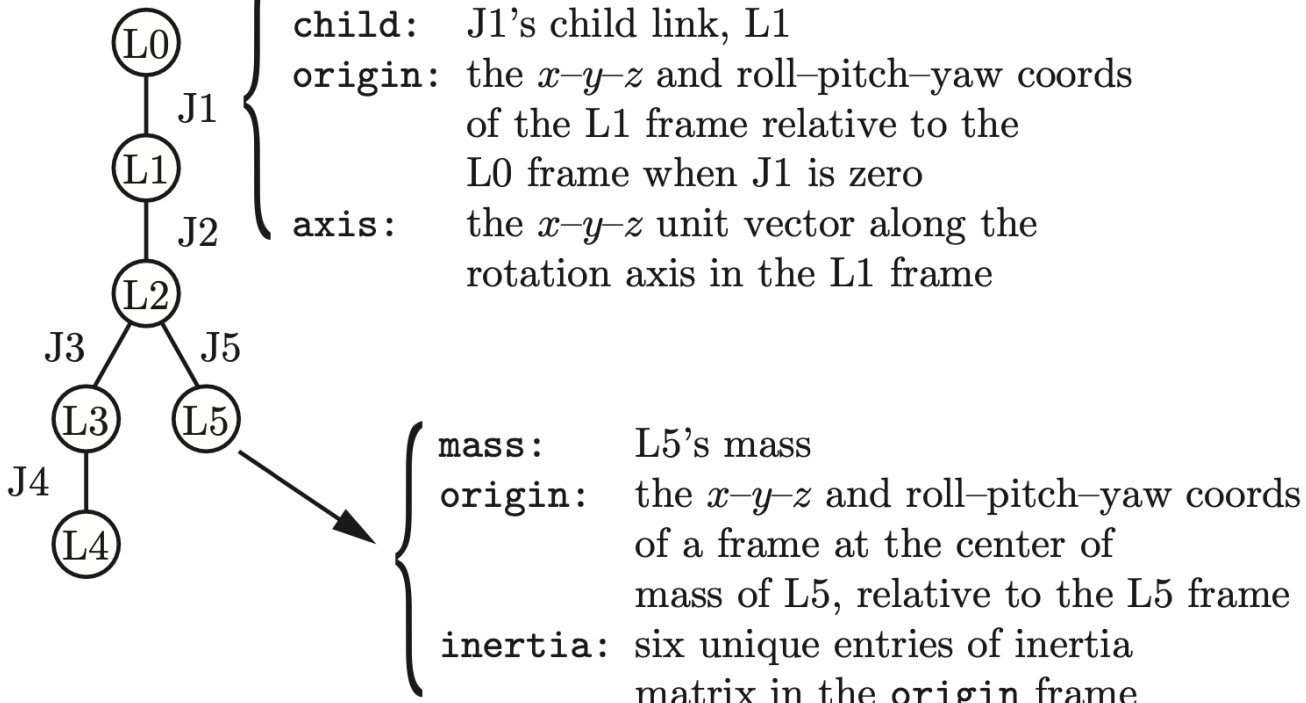


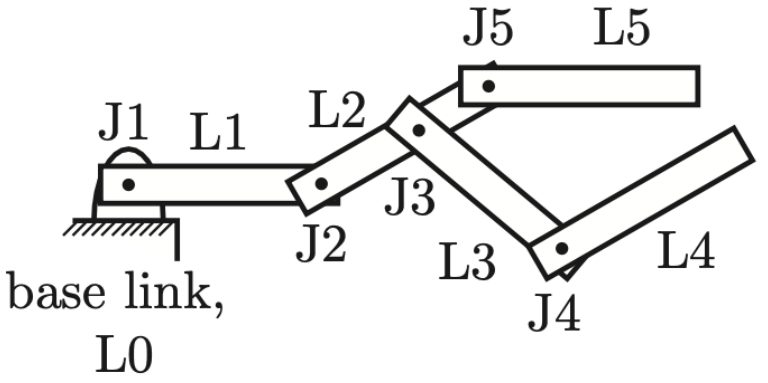
Common Robots are Articulated Objects



Robotics: Modeling, Planning and Control. Siciliano et al.

Articulated Objects are a Tree of Rigid Bodies

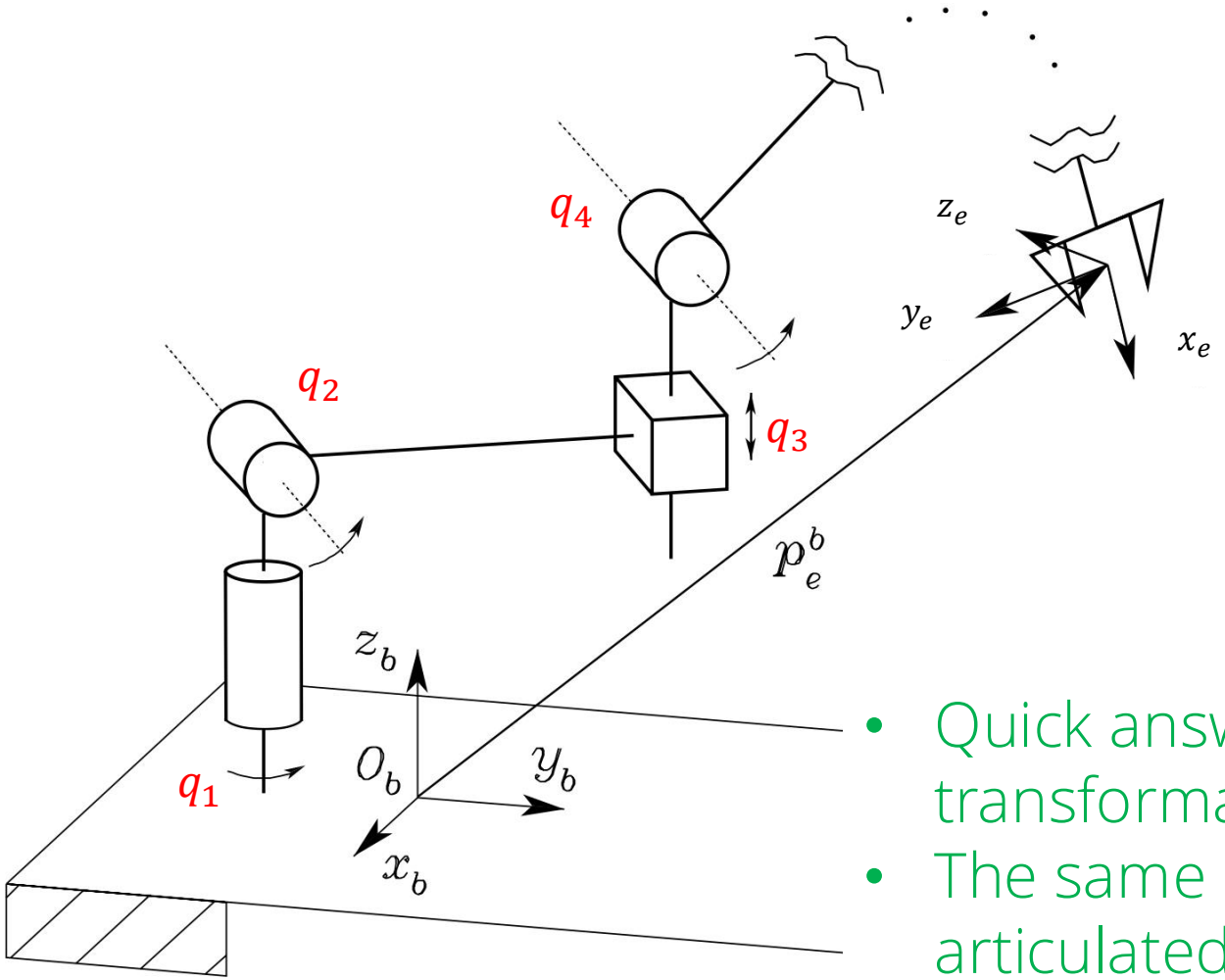




We need to generate:

1. The connectivity graph of parts
2. The geometry and appearance of parts
3. The kinematics of connection (positions, directions and types of joints)

What is the Pose of a Robot's Part given Its Configuration?

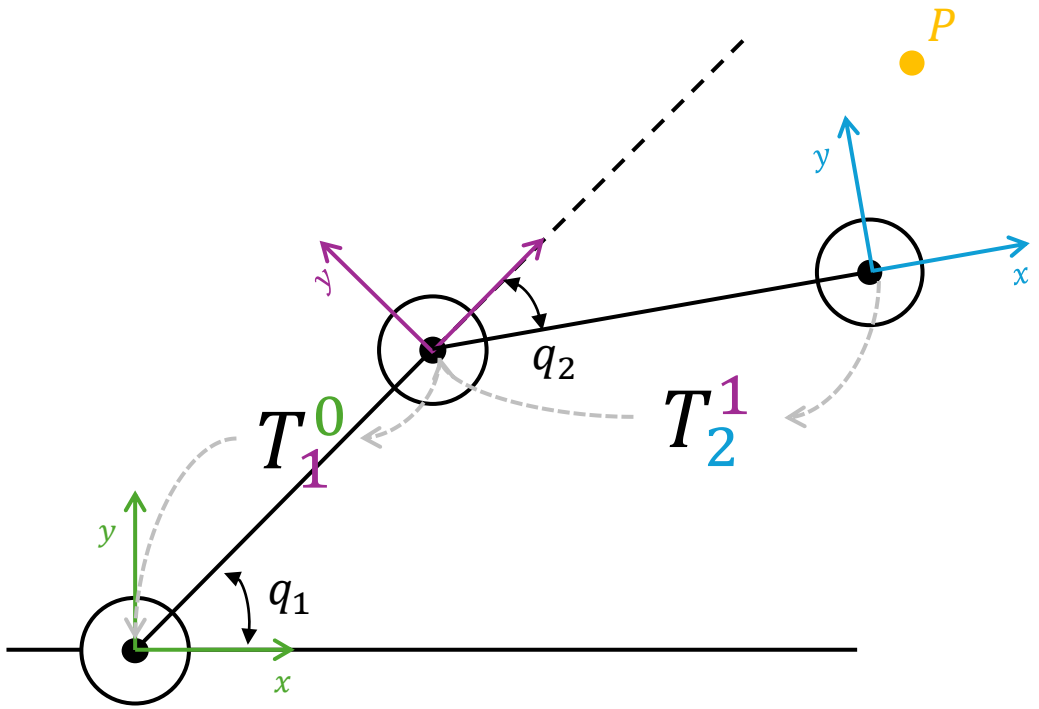


- Quick answer: chaining relative transformation across root parts
- The same idea applies to all articulated objects

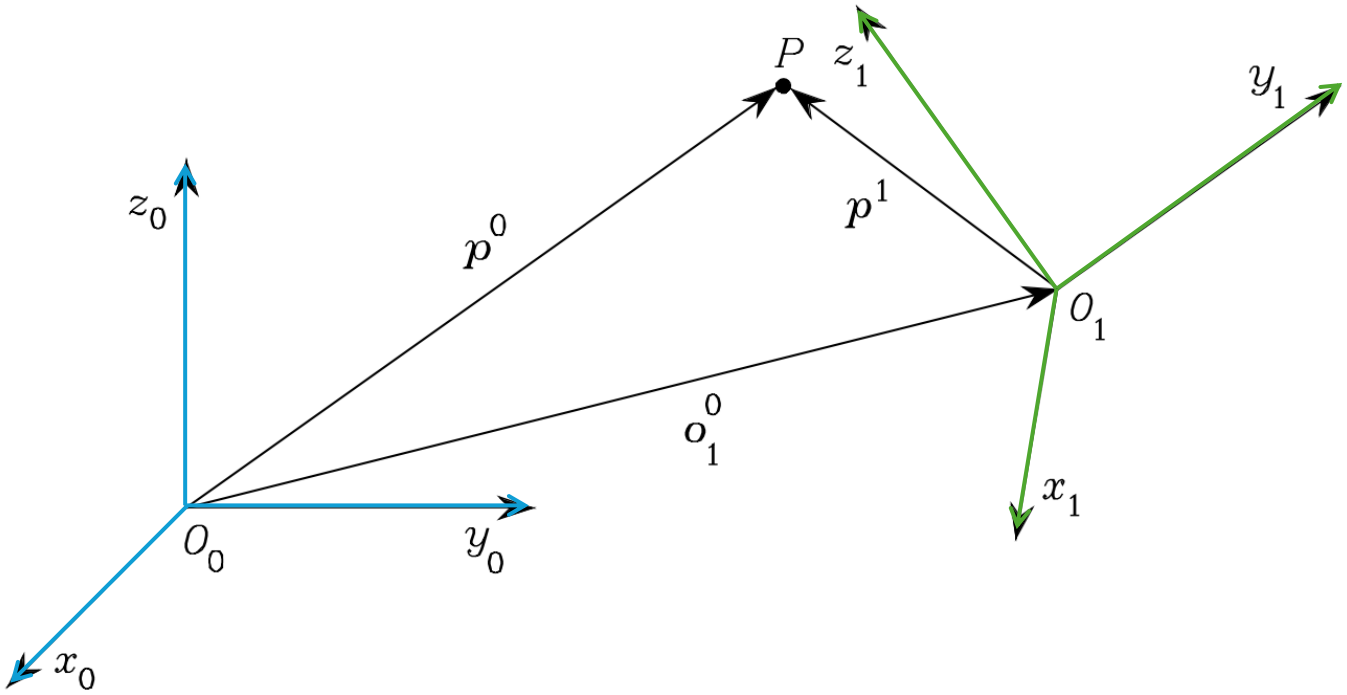
Robotics: Modeling, Planning and Control. Siciliano et al.

How to Calculate the Pose of a Part given the Configuration?

- We can consider each joint applies coordinate transformation



Rigid-Body Transformation



The coordinate of point P in frame O_0

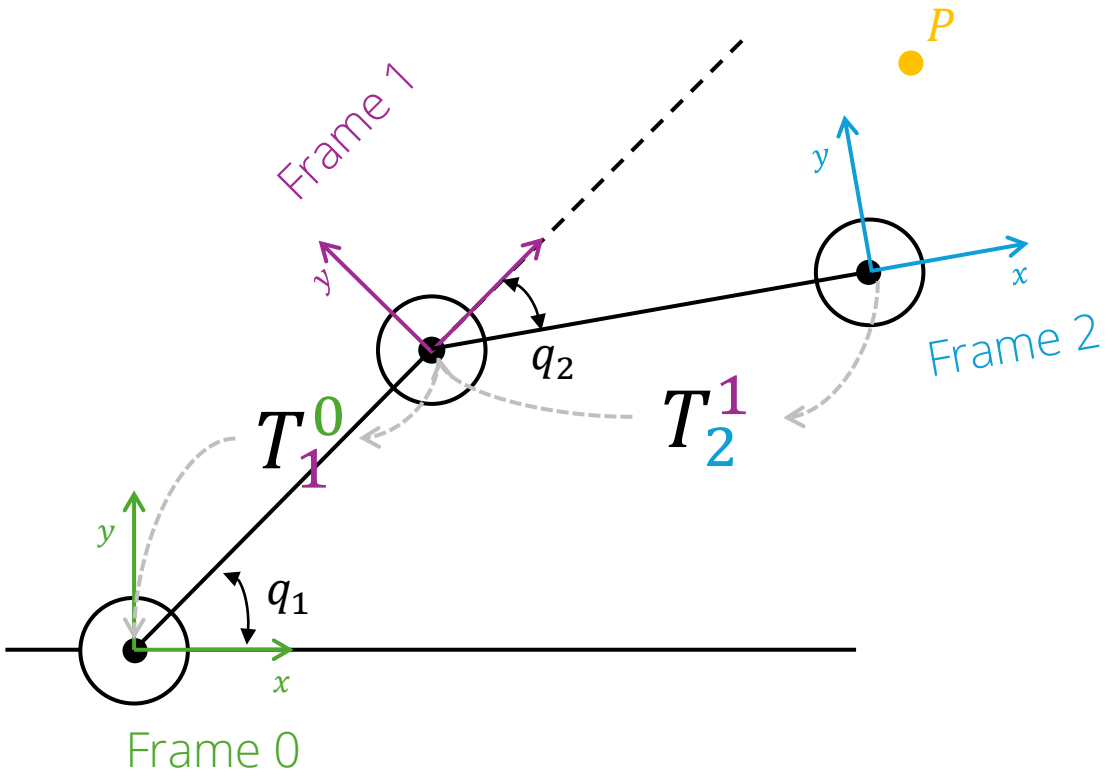
The coordinate of point P in frame O_1

$$\begin{bmatrix} \boldsymbol{p}^0 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_1^0 & \boldsymbol{o}_1^0 \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^1 \\ 1 \end{bmatrix}$$

Spatial Transformation between frame O_1 and O_0

How to Calculate the Pose of a Part given the Configuration?

- We can consider each joint applies coordinate transformation

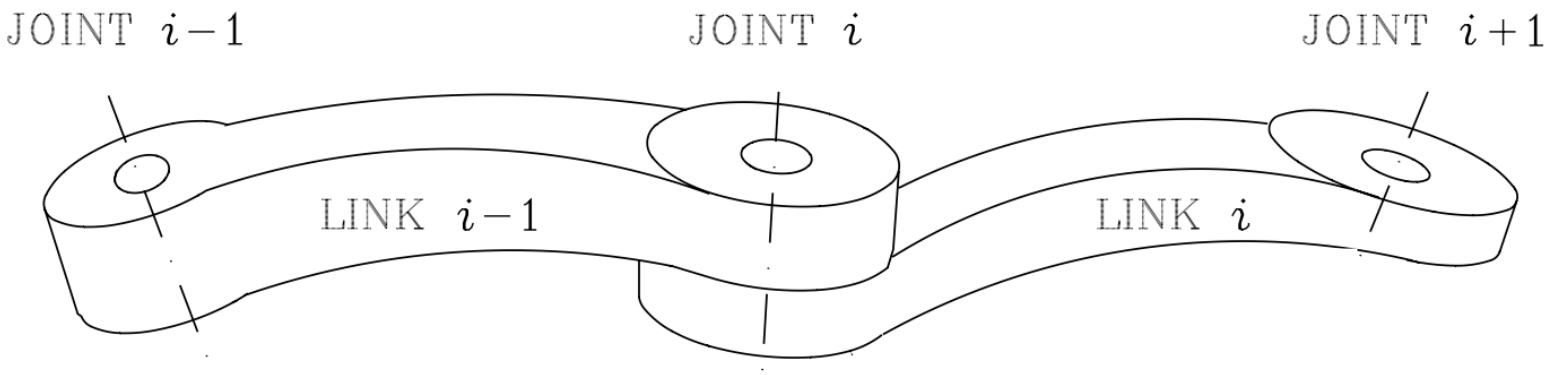


Configuration of Link n

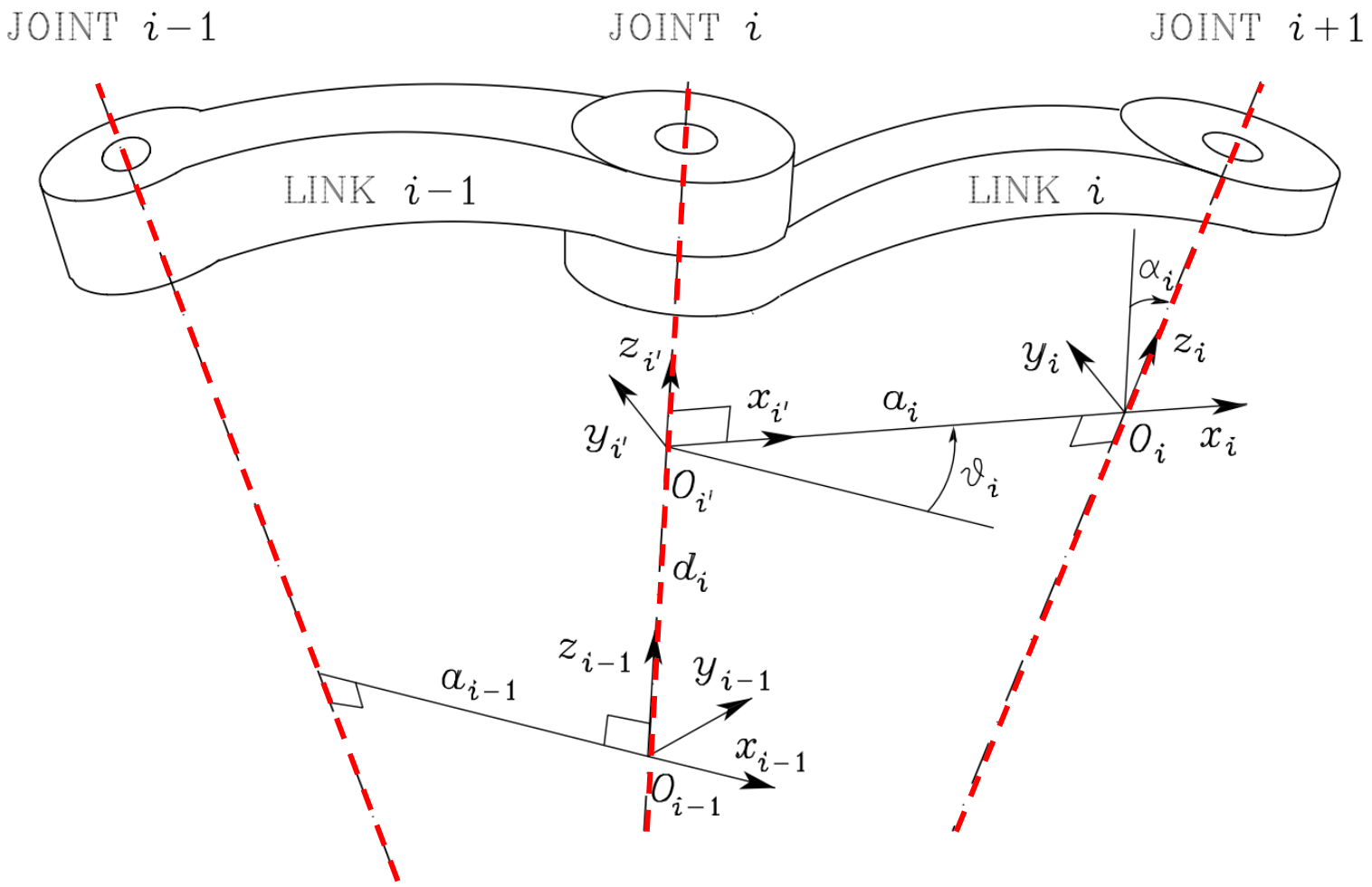
$$T_{\mathrm{n}}^{\mathrm{w}}(\boldsymbol{q}) = T_0^{\mathrm{w}} T_1^0(q_1) \dots T_n^{n-1}(q_n)$$

Coordinate transformation from Frame 0 to the world frame

How to Determine Frames Attached to the Two Links?

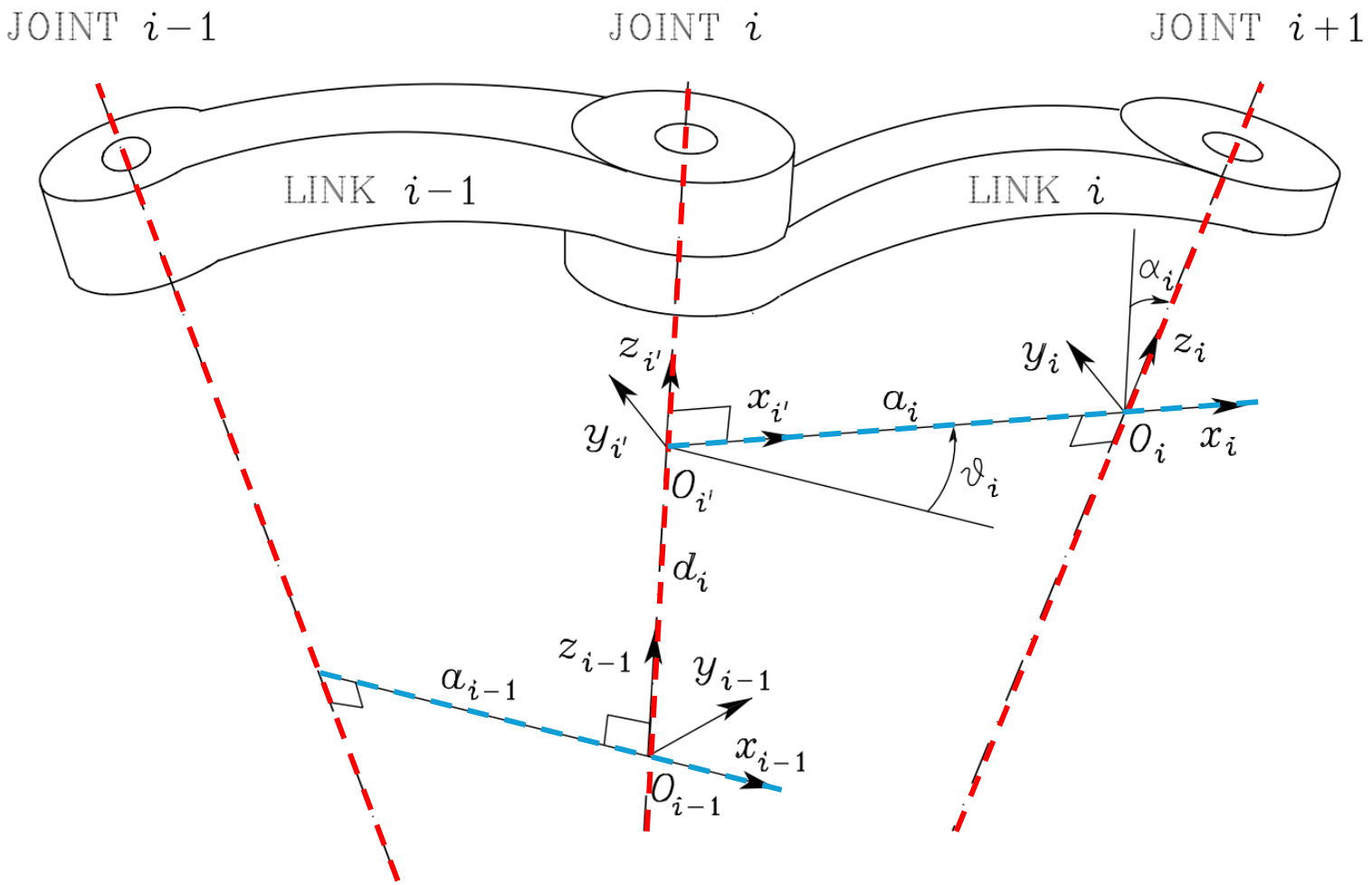


Decide Frames with Denavit–Hartenberg Convention



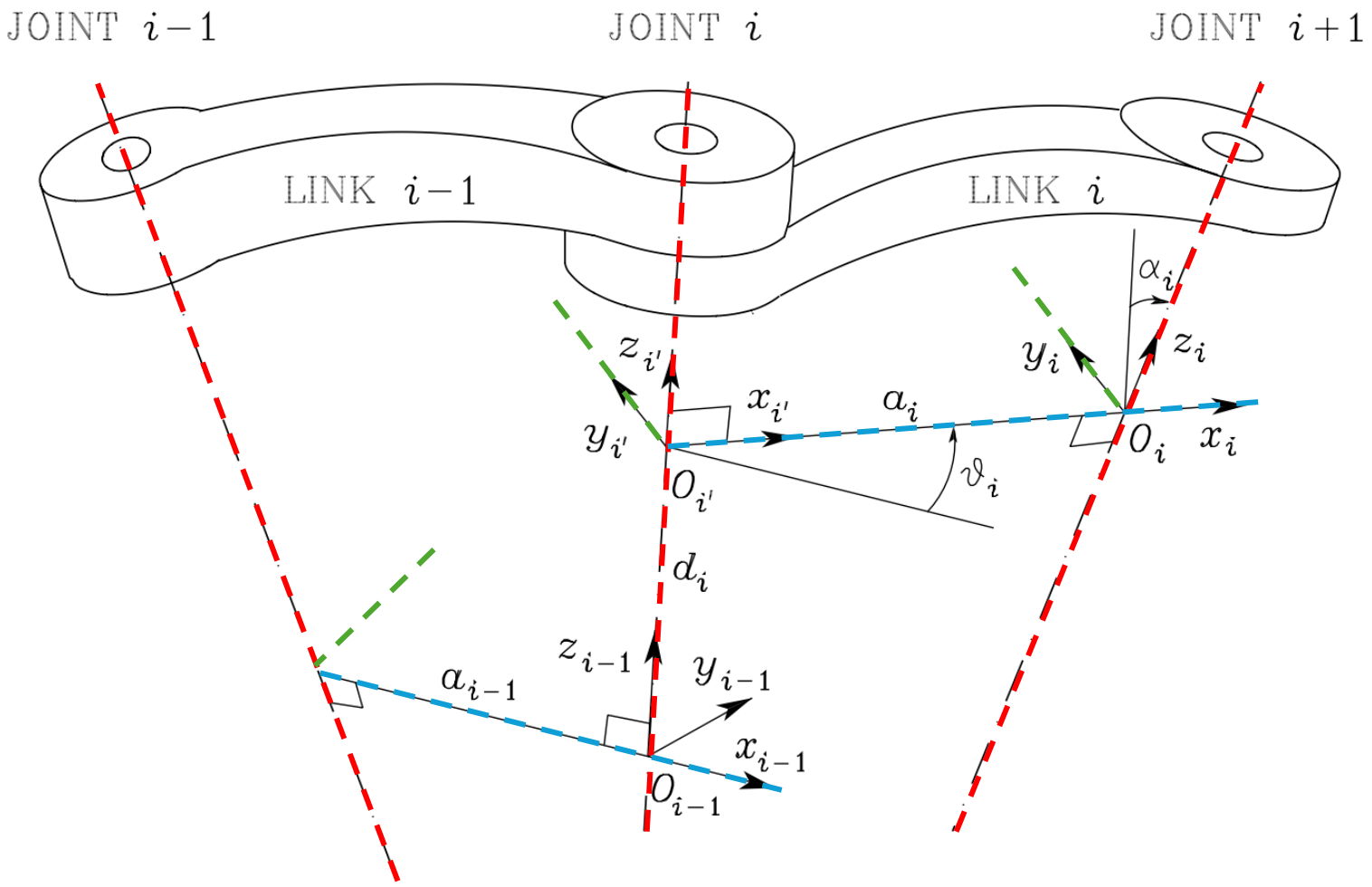
1. Choose axis z_i along the axis of Joint $i+1$

Decide Frames with Denavit–Hartenberg Convention



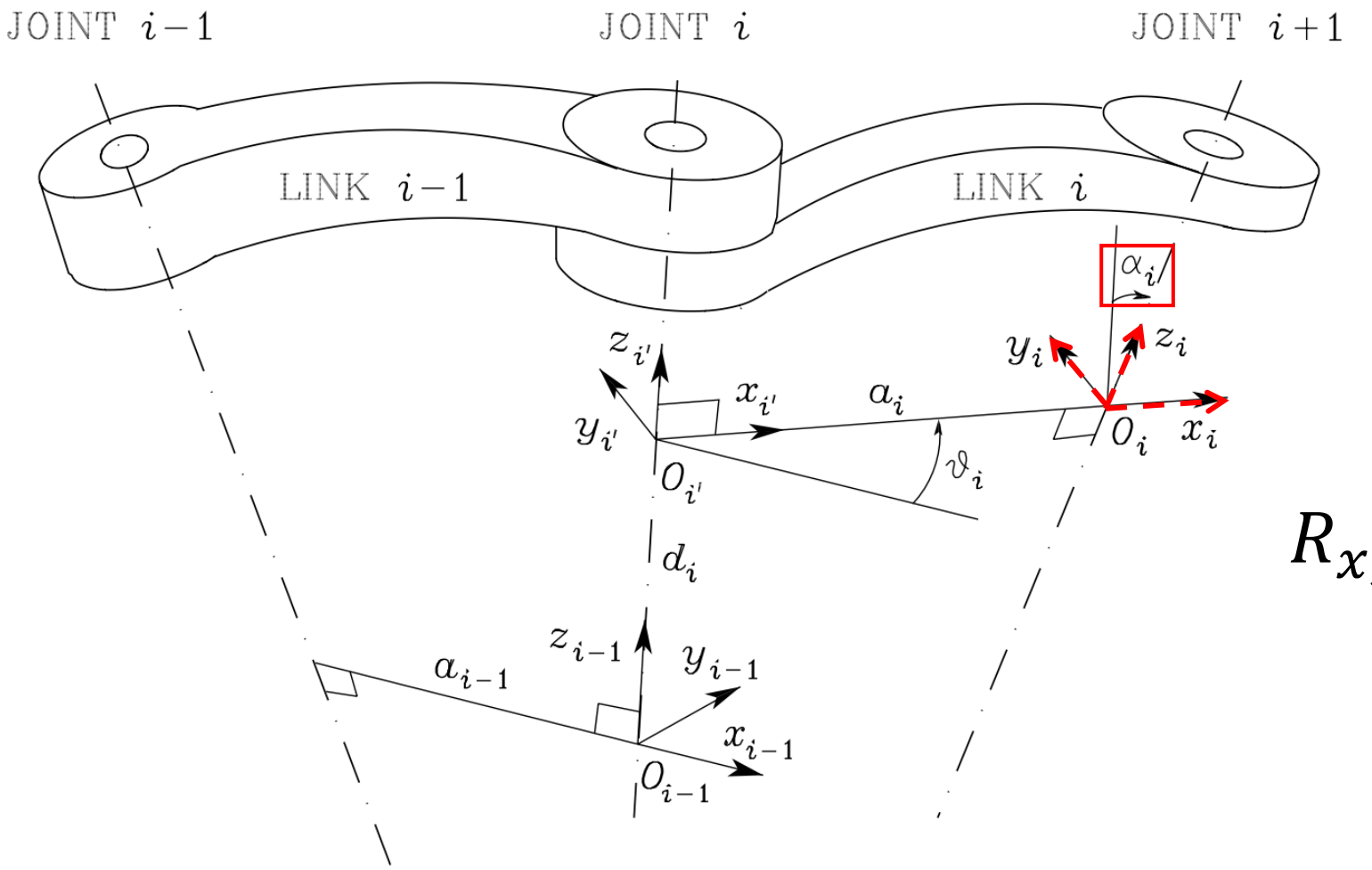
1. Choose axis z_i along the axis of Joint $i+1$
2. Choose axis x_i along the direction of the common normal (vector of minimum distance between axis z_i and z_{i-1})

Decide Frames with Denavit–Hartenberg Convention



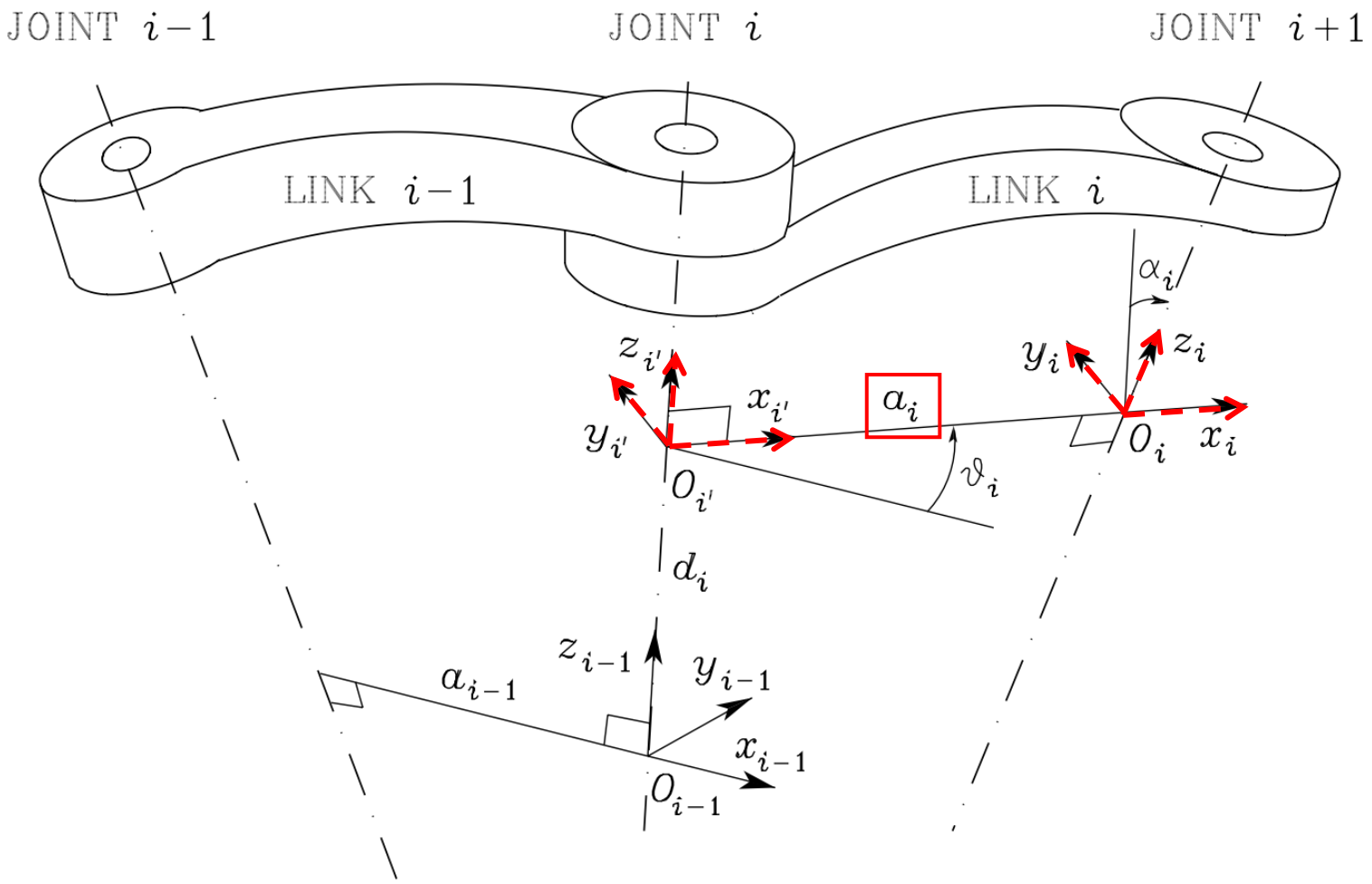
1. Choose axis z_i along the axis of Joint $i+1$
2. Choose axis x_i along the direction of the common normal (vector of minimum distance between axis z_i and z_{i-1})
3. Axis y_i is decided by right-hand rule

What are the Spatial Transformations under Denavit–Hartenberg Convention?



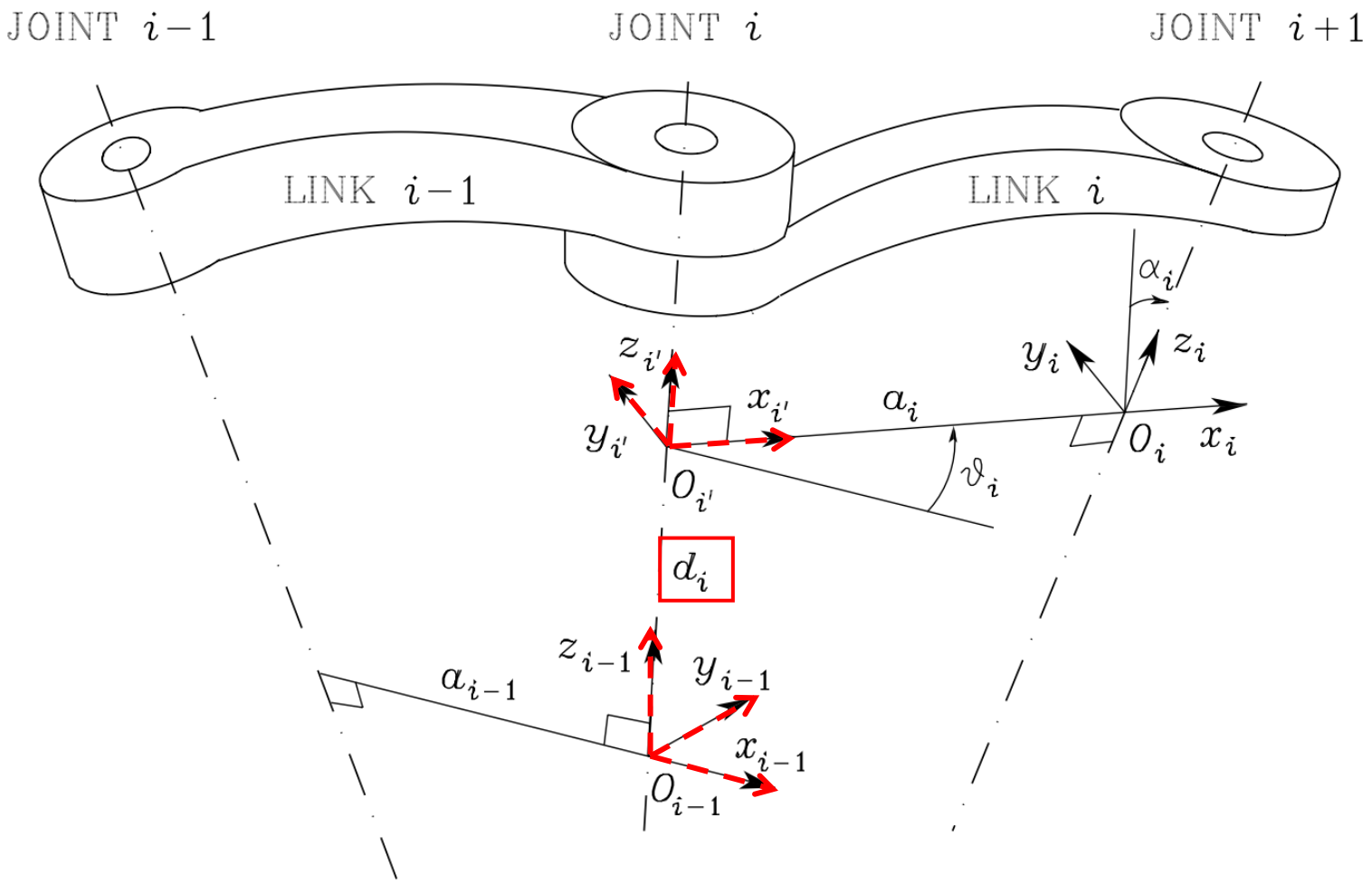
$$R_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

What are the Spatial Transformations under Denavit–Hartenberg Convention?



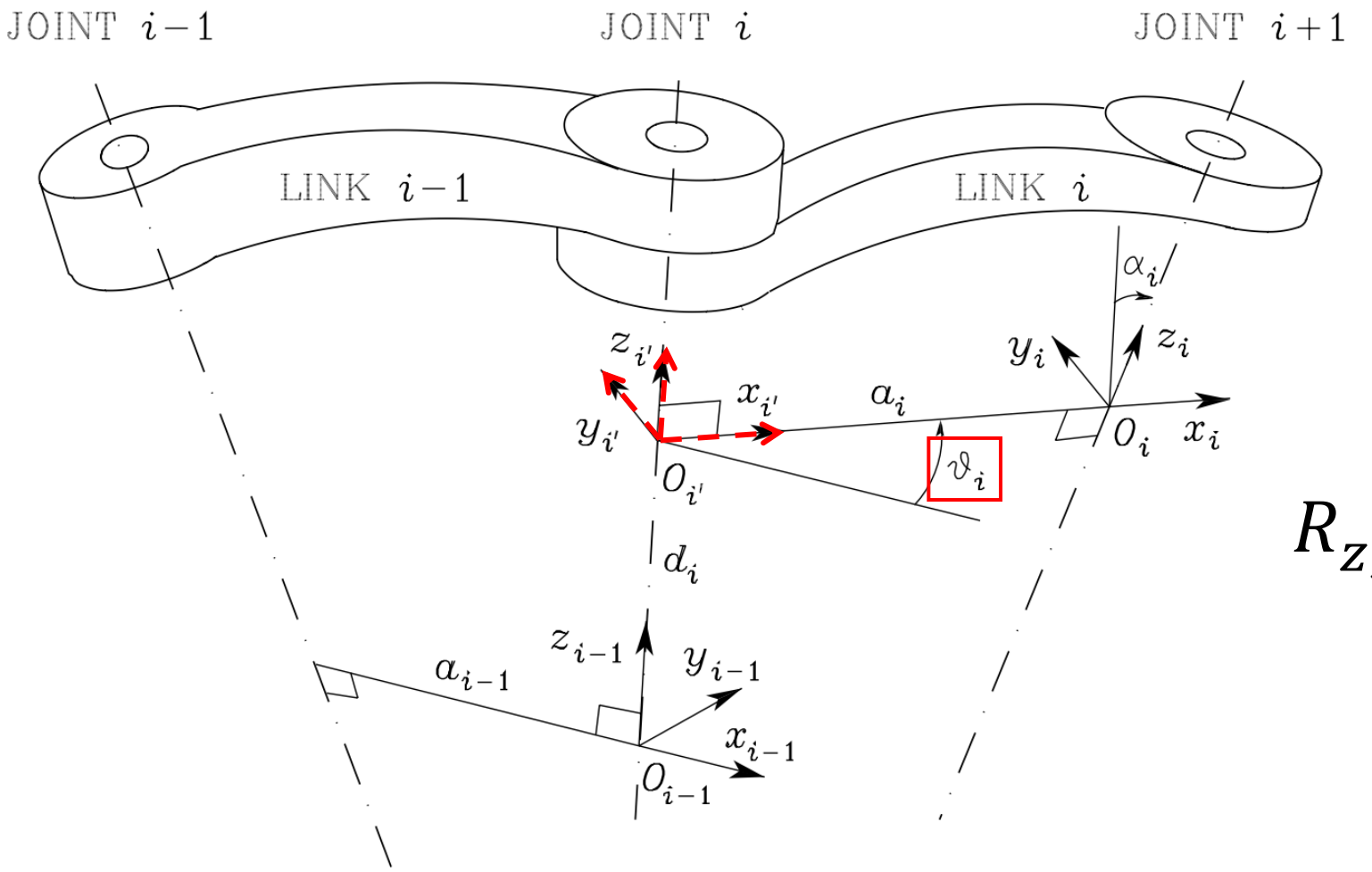
$$T_i^{i'} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

What are the Spatial Transformations under Denavit–Hartenberg Convention?



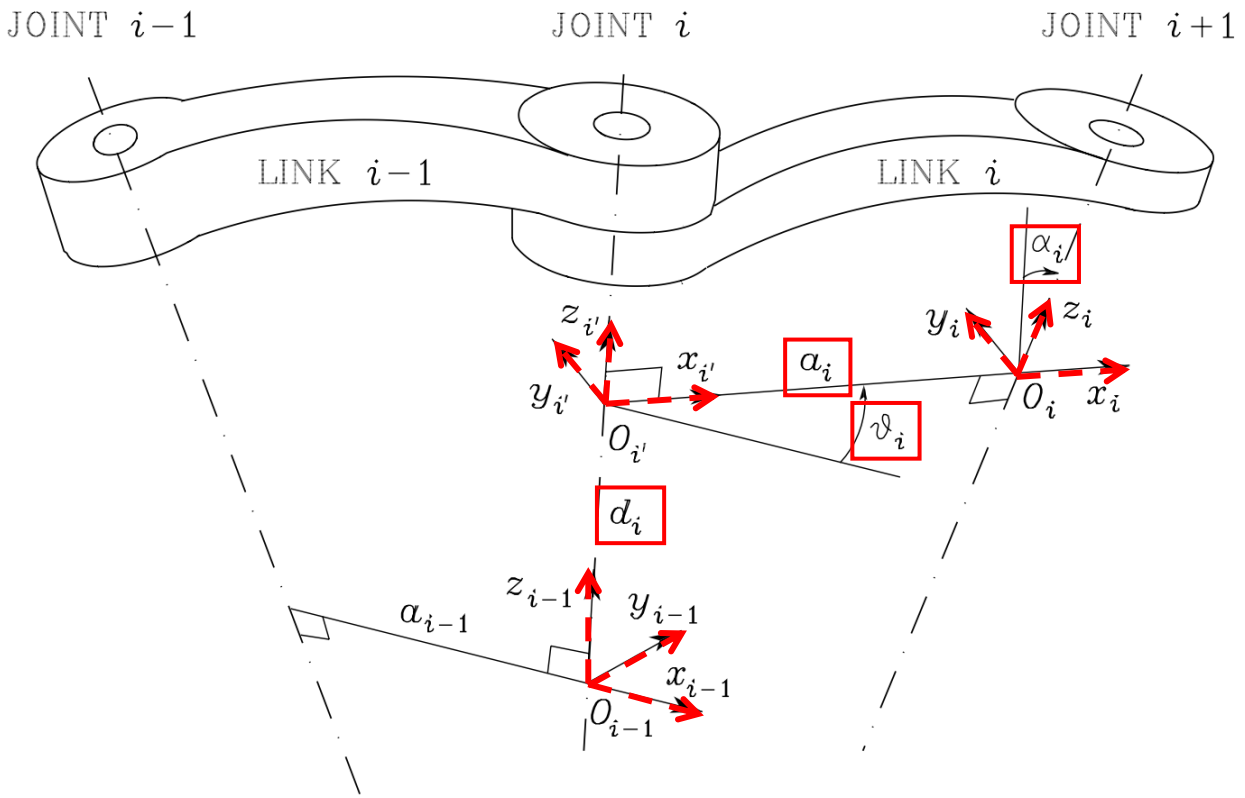
$$T_{i'}^{i-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

What are the Spatial Transformations under Denavit–Hartenberg Convention?



$$R_{z,\vartheta_i} = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i} & 0 & 0 \\ s_{\vartheta_i} & c_{\vartheta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

What are the Spatial Transformations under Denavit–Hartenberg Convention?



$$A_i^{i-1} = R_{z,\vartheta_i} T_{i'}^{i-1} T_i^{i'} R_{x,\alpha_i}$$

$$= \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i} c_{\alpha_i} & s_{\vartheta_i} s_{\alpha_i} & a_i c_{\vartheta_i} \\ s_{\vartheta_i} & c_{\vartheta_i} c_{\alpha_i} & -c_{\vartheta_i} s_{\alpha_i} & a_i s_{\vartheta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward Kinematics: Calculate the Pose of a Part given the Configuration

- We known:

Transformation from frame i to frame $i-1$

$$A_i^{i-1} = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i}c_{\alpha_i} & s_{\vartheta_i}s_{\alpha_i} & a_i c_{\vartheta_i} \\ s_{\vartheta_i} & c_{\vartheta_i}c_{\alpha_i} & -c_{\vartheta_i}s_{\alpha_i} & a_i s_{\vartheta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation from frame $\mathbf{0}$ to the world frame

Transformation from the frame $\mathbf{n}$ to the world frame

Configuration of joint n

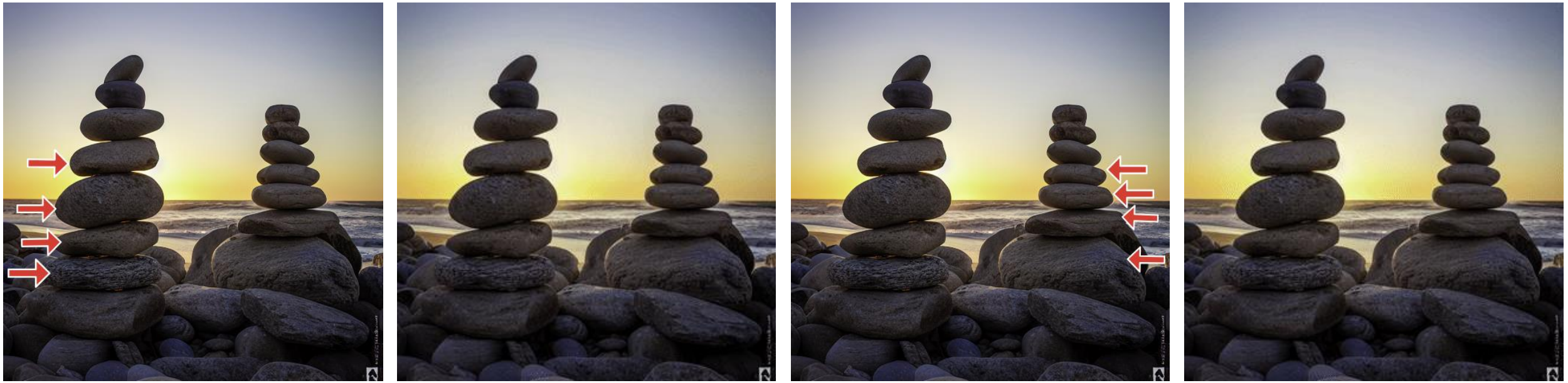
$$A_{\mathrm{n}}^{\mathrm{w}}(\boldsymbol{q}) = A_0^{\mathrm{w}} A_1^0(q_1) \dots A_n^{n-1}(q_n)$$

$$\boldsymbol{p}_{\mathrm{n}}^{\mathrm{w}} = A_0^{\mathrm{w}} A_1^0(q_1) \dots A_n^{n-1}(q_n)\boldsymbol{p}$$

Content

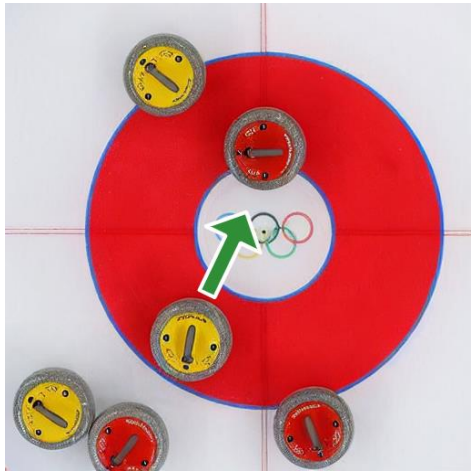
- Rigid-body Modeling
  - ➢Physical law of rigid bodies
  - ➢Formulation of articulated objects
- Dynamic Modeling of Rigid Bodies
  - ➢Video generation of rigid-body motions
  - ➢Generation of 3D articulated objects

Physically Accurate Video Generator



PhysGen: Rigid-Body Physics-Grounded Image-to-Video Generation. Liu et al.

Idea: Integrate Video Generative Model with a Rigid-Body Motion Simulator

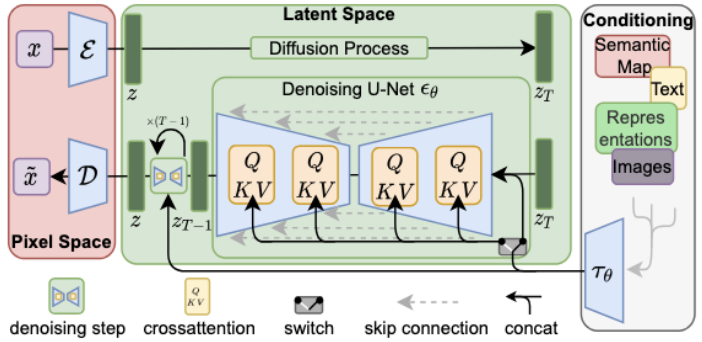


Rigid-body physic law:

$$\frac{d}{dt}\mathbf{q}(t) = \frac{d}{dt}\begin{bmatrix}\mathbf{t}(t)\\ \mathbf{R}(t)\\ \boldsymbol{\nu}(t)\\ \boldsymbol{\omega}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{v}(t)\\ \boldsymbol{\omega}(t)\times\mathbf{R}(t)\\ \frac{\mathbf{F}(t)}{M}\\ \mathbf{I}(t)^{-1}(\boldsymbol{\tau}-\boldsymbol{\omega}(t)\times\mathbf{I}(t)\boldsymbol{\omega}(t))\end{bmatrix}$$

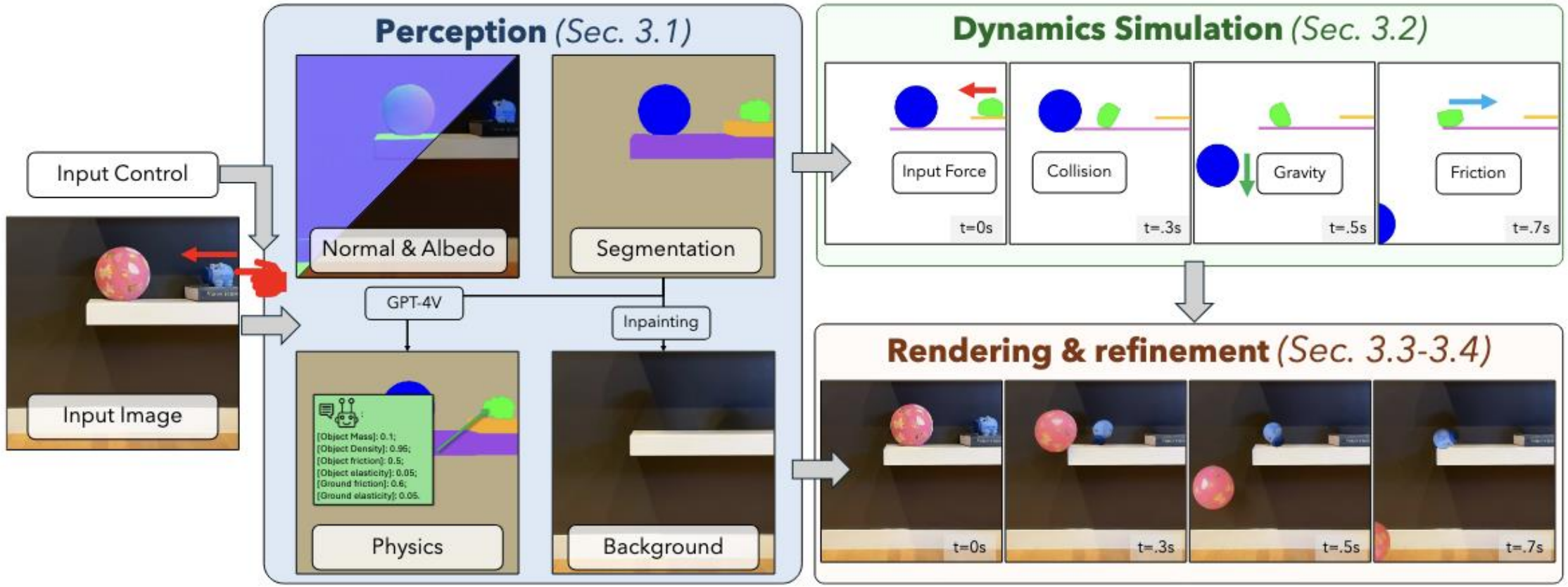
Simulation:

$$\mathbf{q}(t) = \mathbf{q}(0) + \int_0^t \frac{d}{dt}\mathbf{q}(t)dt = \mathbf{q}(0) + \sum_{i=1}^{T}\frac{d}{dt}\mathbf{q}(t)|_{t=t_i}\Delta_t$$



PhysGen: Rigid-Body Physics-Grounded Image-to-Video Generation. Liu et al.

PhysGen: Rigid-Body Physics-Grounded Image-to-Video Generation



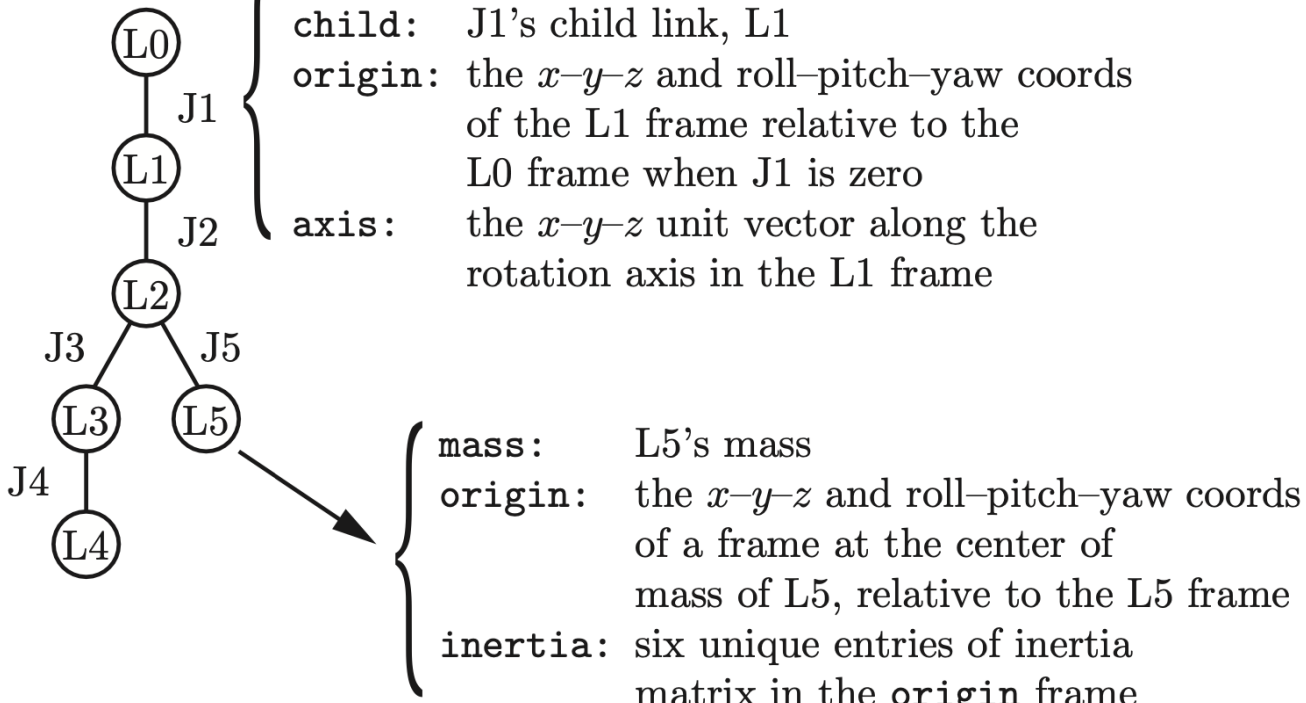
Results: Interactive Video Generation

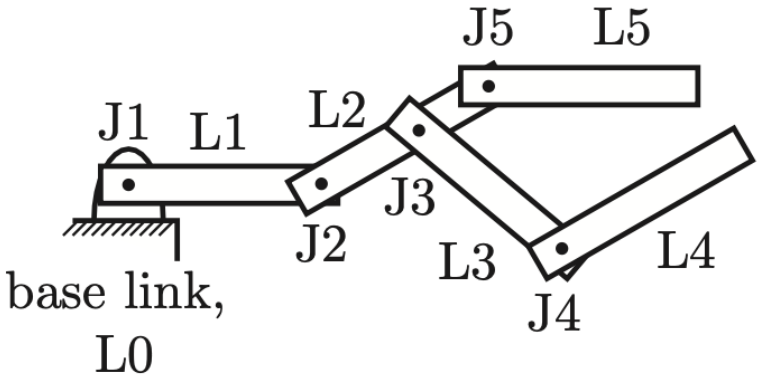
- Check interactive demo

Content

- Rigid-body Modeling
  - ➢Physical law of rigid bodies
  - ➢Formulation of articulated objects
- Dynamic Modeling of Rigid Bodies
  - ➢Video generation of rigid-body motions
  - ➢Generation of 3D articulated objects

Articulated Objects are a Tree of Rigid Bodies





We need to generate:

1. The connectivity graph of parts
2. The geometry and appearance of parts
3. The kinematics of connection (positions, directions and types of joints)

An Easier Problem: Reconstruct 3D Articulated Objects from an Input Image

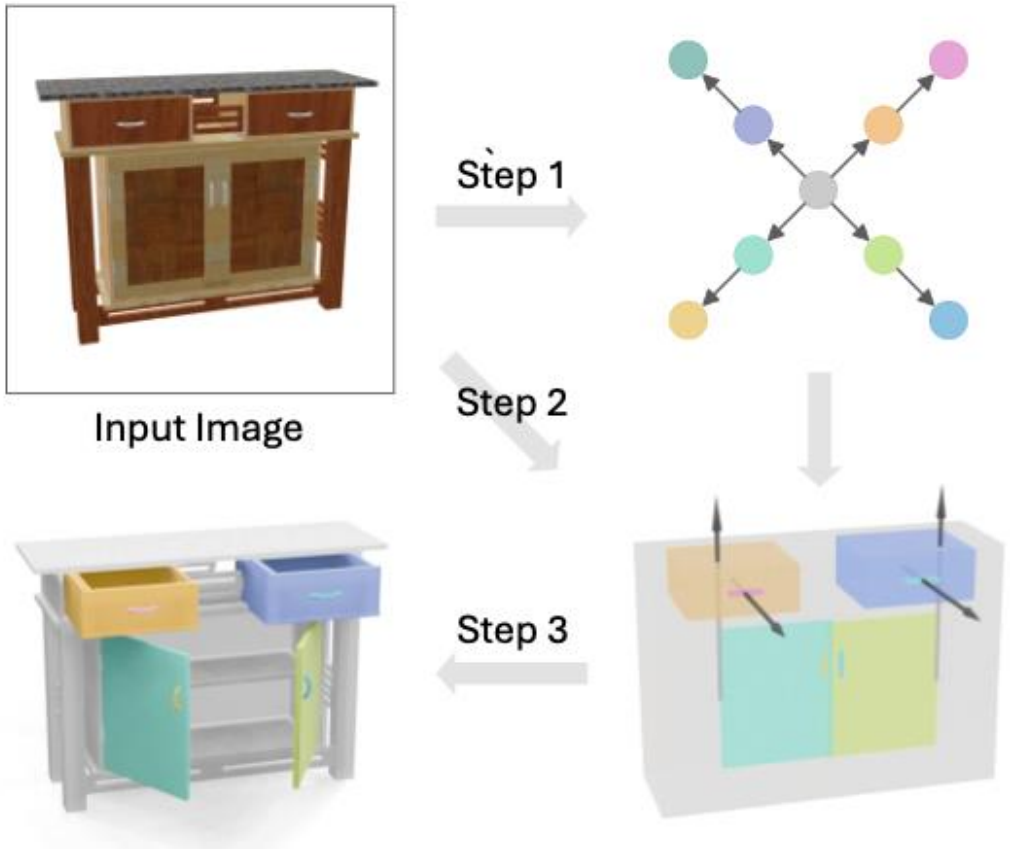
reconstruct

We need to reconstruct:

1. The connectivity graph of parts
2. The geometry and appearance of parts
3. The kinematics of connection (positions, directions and types of joints)

Image source: FreeArt3D: Training-Free Articulated Object Generation using 3D Diffusion. Chen et al.

Idea: Breakdown Reconstruction Pipeline

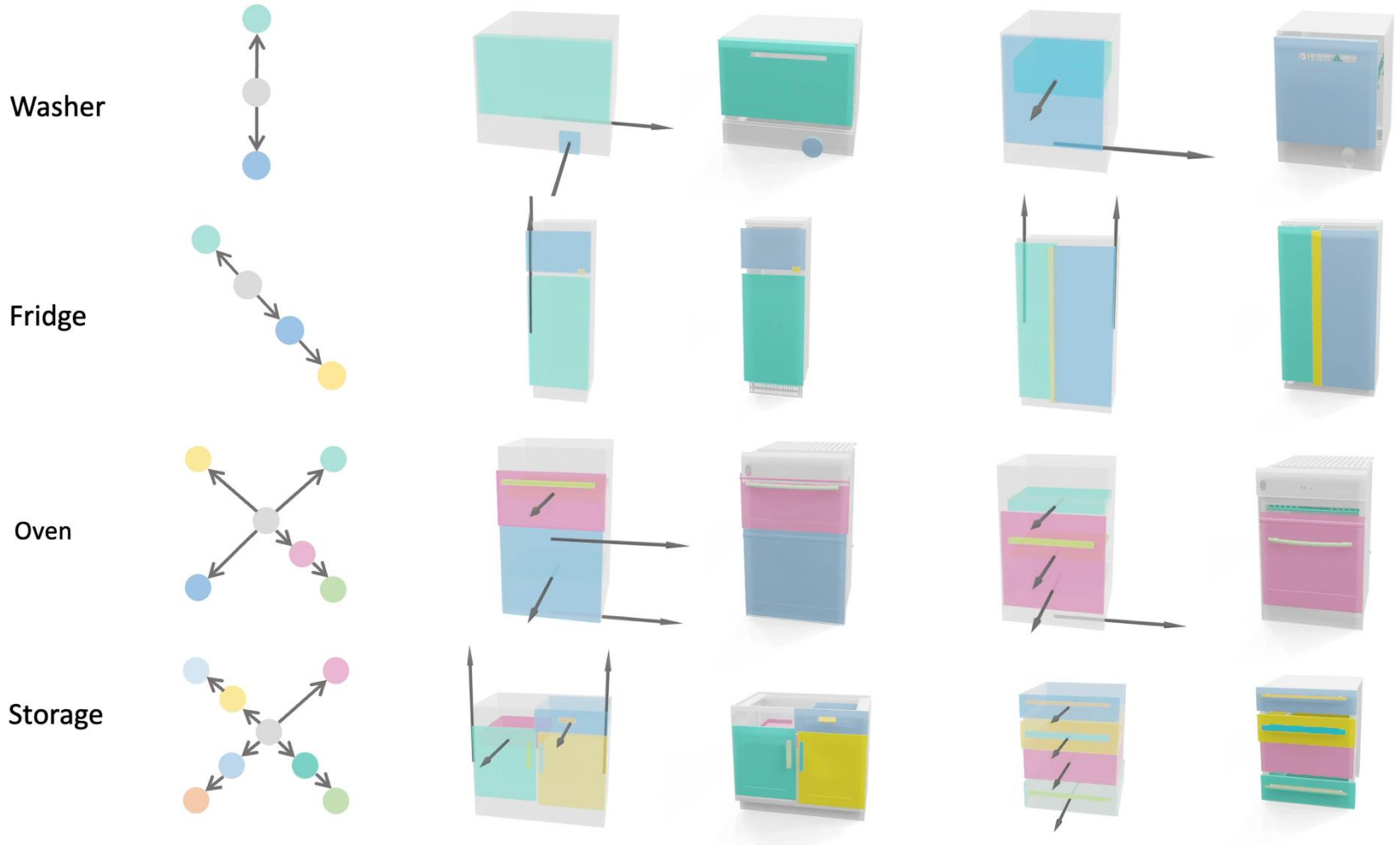


Reconstruction pipeline:

1. Infer the connectivity graph of parts from input image
2. Infer the kinematics of connection (positions, directions and types of joints)
3. Reconstruct the geometry and appearance of parts

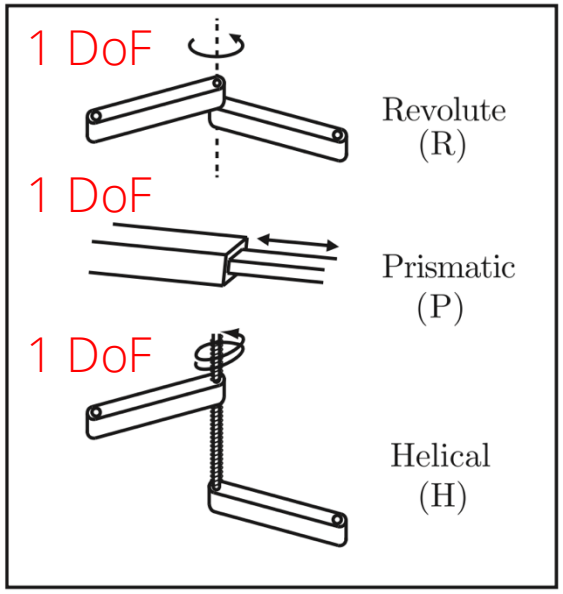
SINGAPO: Single Image Controlled Generation of Articulated Parts in Objects. Liu et al.

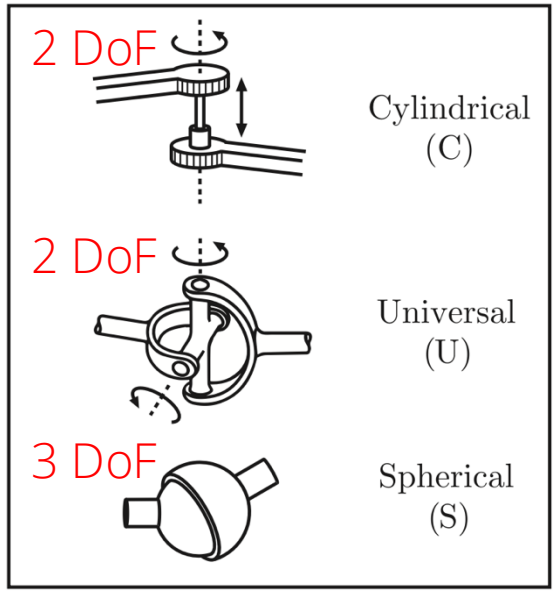
How to Represent the Connective Graph?



CAGE: Controllable Articulation GEneration. Liu et al.

How to Represent the Kinematics of Connections?





For simplification, current work only consider a limited set of connections:

- Articulation type: fixed, revolute, prismatic, continuous, and screw joints
- Kinematic parameters:
  - The location and rotation axis
  - The motion range

How to Represent the Geometry and Appearance of Parts?

For each part, we need to denote:

- Size of each part with bounding box
- Geometry and appearance of each part with structured latents or object code

Image source: OmniPart: Part-Aware 3D Generation with Semantic Decoupling and Structural Cohesion. Yang et al.

SINGAPO: Single Image Controlled Generation of Articulated Parts in Objects

Jiayi Liu[1], Denys Iliash[1], Angel X. Chang[1,2], Manolis Savva[1], Ali Mahdavi-Amiri[1]
[1]Simon Fraser University, [2]Canada-CIFAR AI Chair, Amii

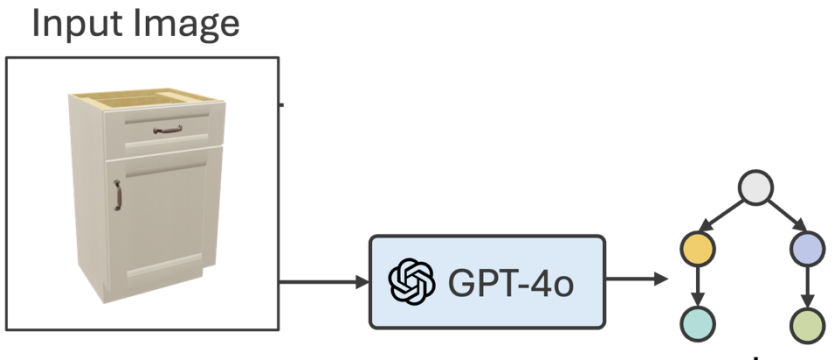
ICLR 2025

arXiv Code

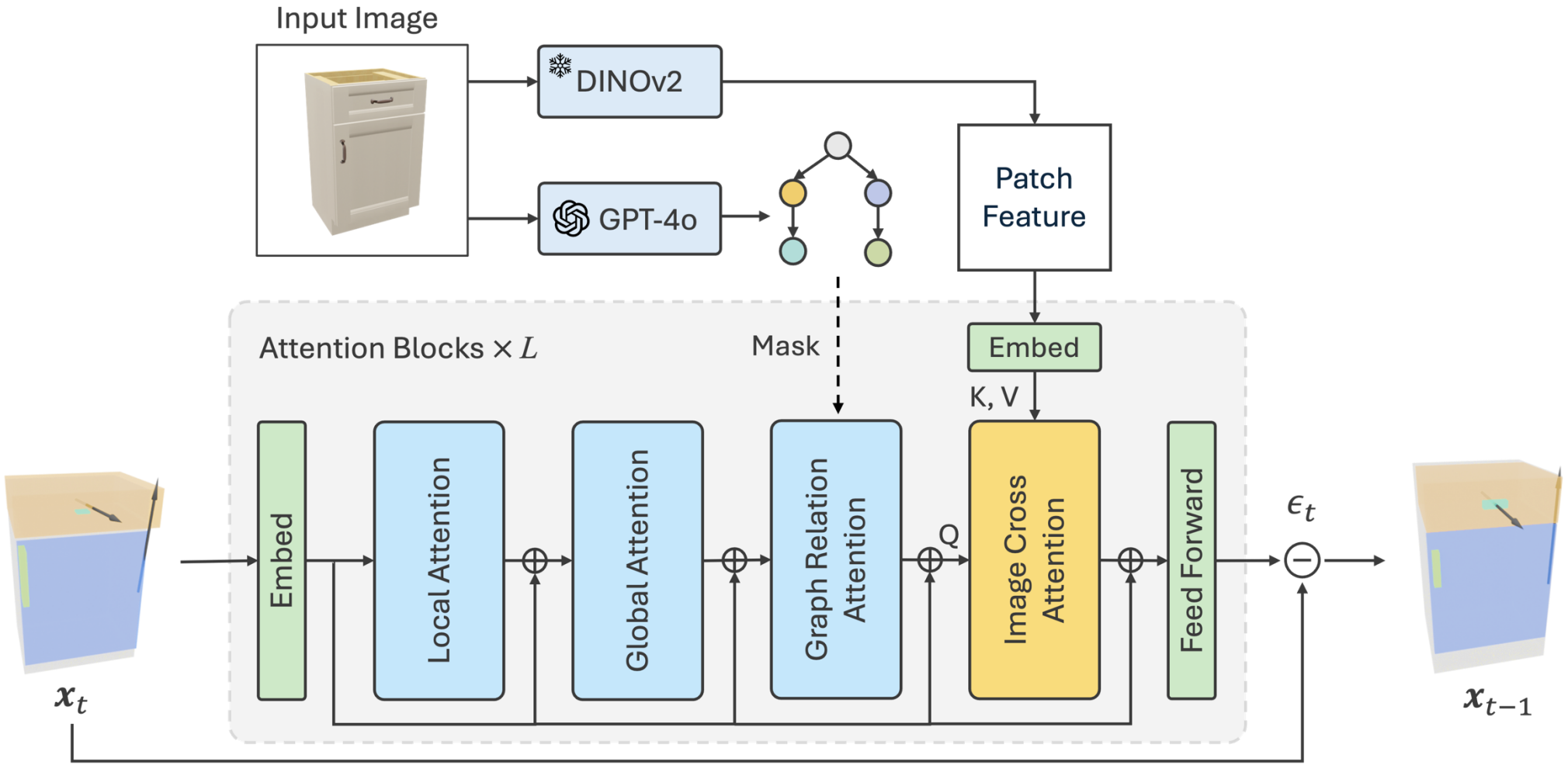
Reconstruction pipeline:

1. Infer the connectivity graph of parts from input image with VLMs
2. Infer the kinematics of connection (positions, directions and types of joints) by learning a transformer
3. Reconstruct the geometry and appearance of parts by retrieving 3D assets from a dataset

Step 1: Infer the Connectivity Graph of Parts from Input Image with VLMs

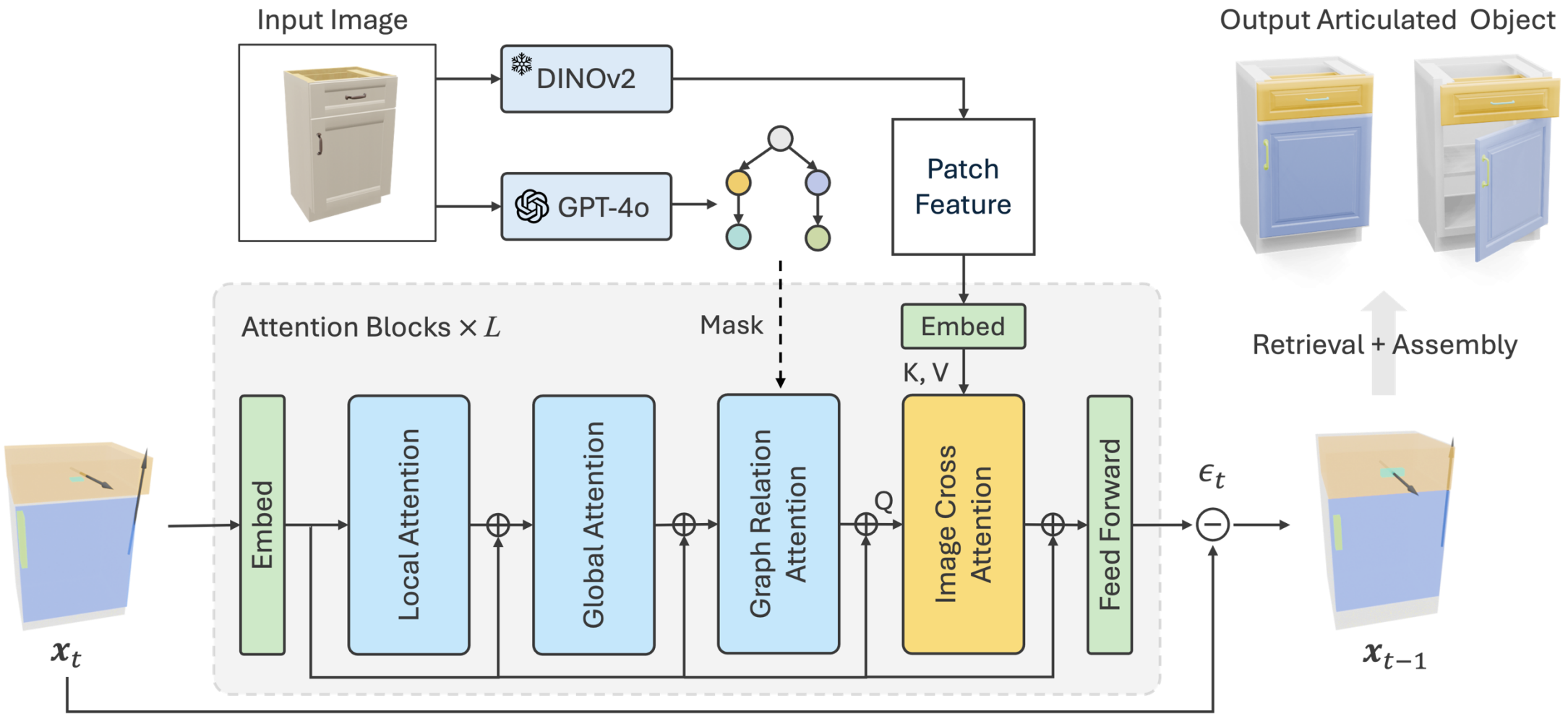


Step 2: Infer the Kinematics of Connection by Learning a Transformer

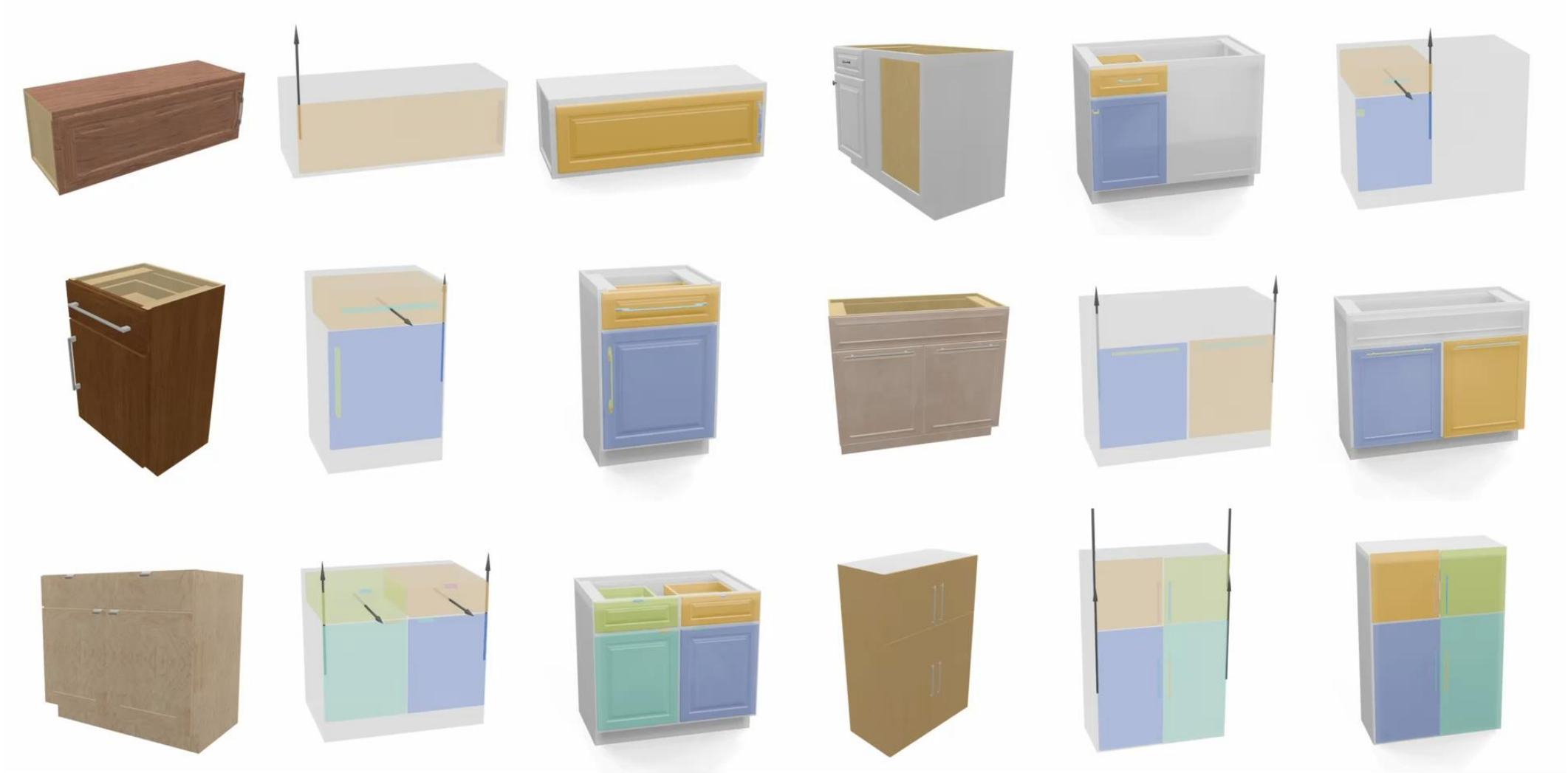


SINGAPO: Single Image Controlled Generation of Articulated Parts in Objects. Liu et al.

Step 3: Retrieve and Assemble 3D Assets from a Dataset



SINGAPO: Single Image Controlled Generation of Articulated Parts in Objects. Liu et al.



SINGAPO: Single Image Controlled Generation of Articulated Parts in Objects. Liu et al.

ArtFormer: Controllable Generation of Diverse 3D Articulated Objects

Jiayi Su[1,*,†] Youhe Feng[2,*] Zheng Li[4] Jinhua Song[1] Yangfan He[5,6] Botao Ren[3] Botian Xu[3,‡]
[1]Xiamen University Malaysia [2]Renmin University of China [3]Tsinghua University
[4]Southern University of Science and Technology [5]University of Minnesota-Twin Cities
[6]Henan RunTai Digital Technology Group Co., Ltd.
* Equal Contribution. †CST2209162@xmu.edu.my ‡btx0424@outlook.com

github.com/ShuYuMo2003/ArtFormer

Reconstruction pipeline:

1. Infer the connectivity graph of parts from input image by learning a transformer
2. Infer the kinematics of connection (positions, directions and types of joints) by learning a transformer
3. Reconstruct the geometry and appearance of parts by generating structured latents

ArtFormer: Controllable Generation of Diverse 3D Articulated Objects. Su et al.

Idea: Auto-Regressive Models Naturally Generate Tree of Data

Example: Beam search with Large Language Models

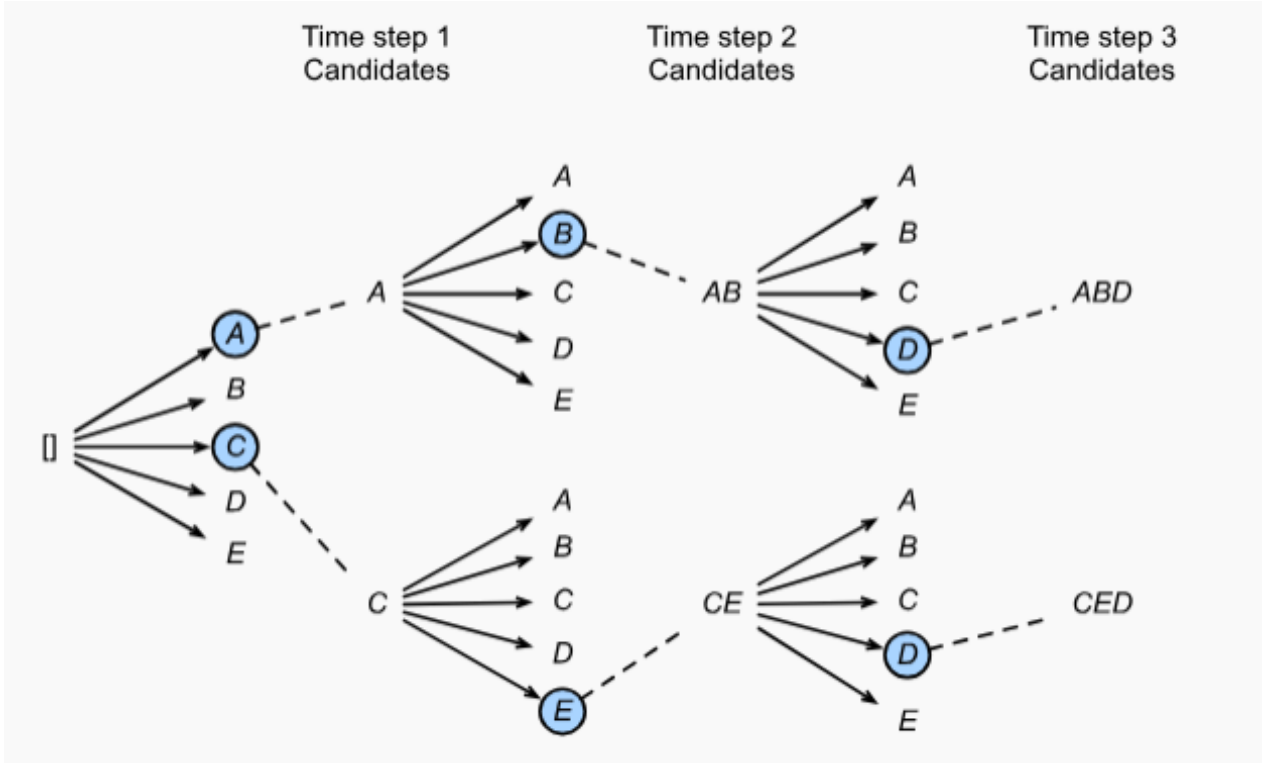
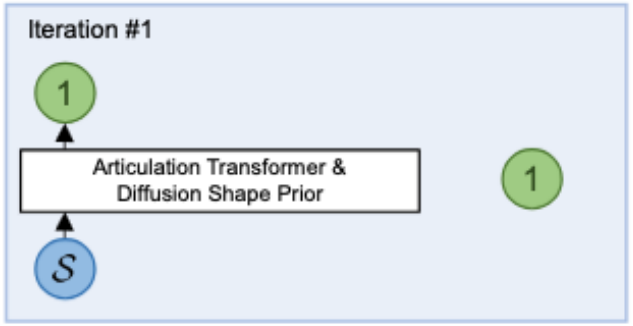
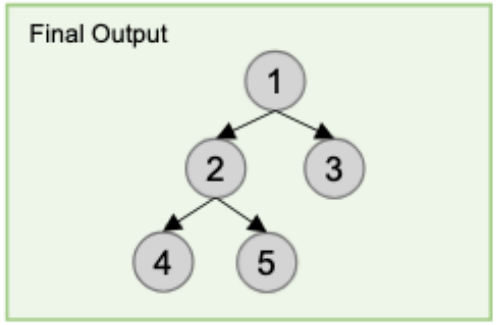


Image source: https://d2l.ai/chapter_recurrent-modern/beam-search.html

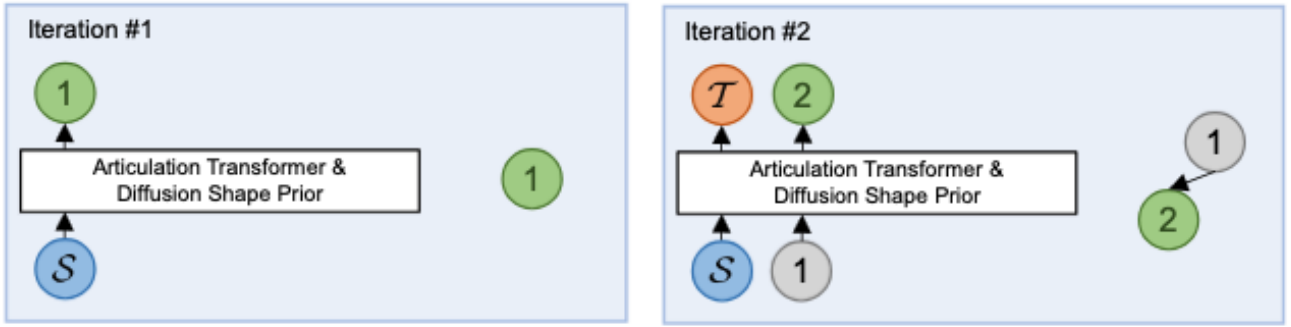
Idea: Sample Tree of Data with Transformers

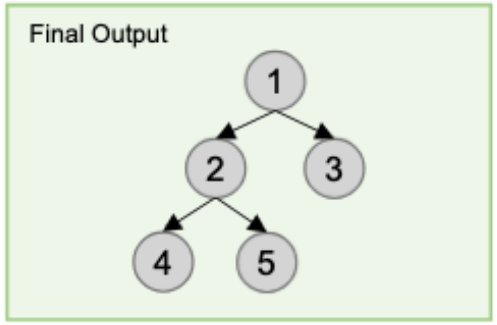




ArtFormer: Controllable Generation of Diverse 3D Articulated Objects. Su et al.

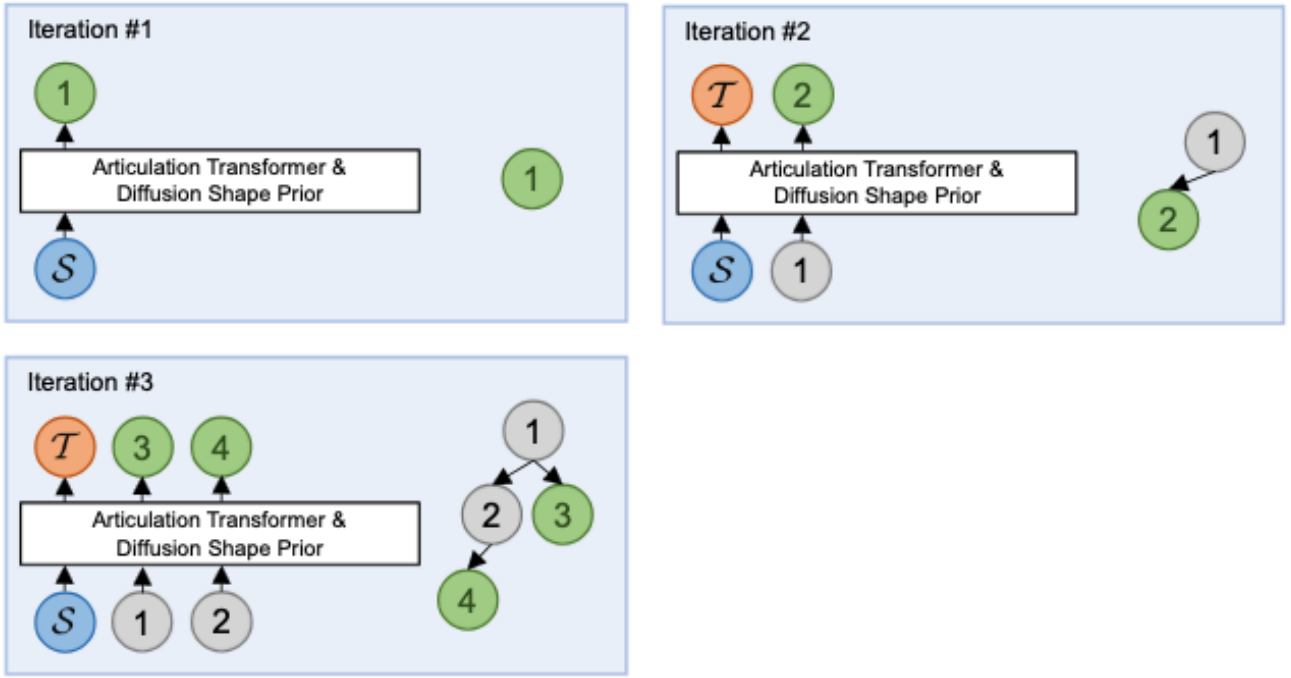
Idea: Sample Tree of Data with Transformers

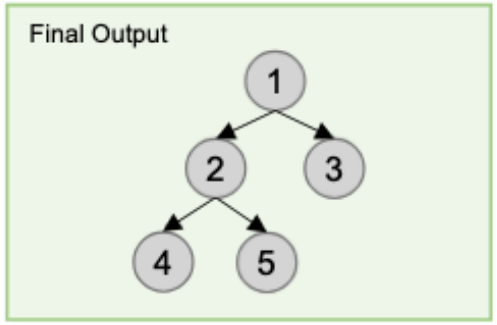




ArtFormer: Controllable Generation of Diverse 3D Articulated Objects. Su et al.

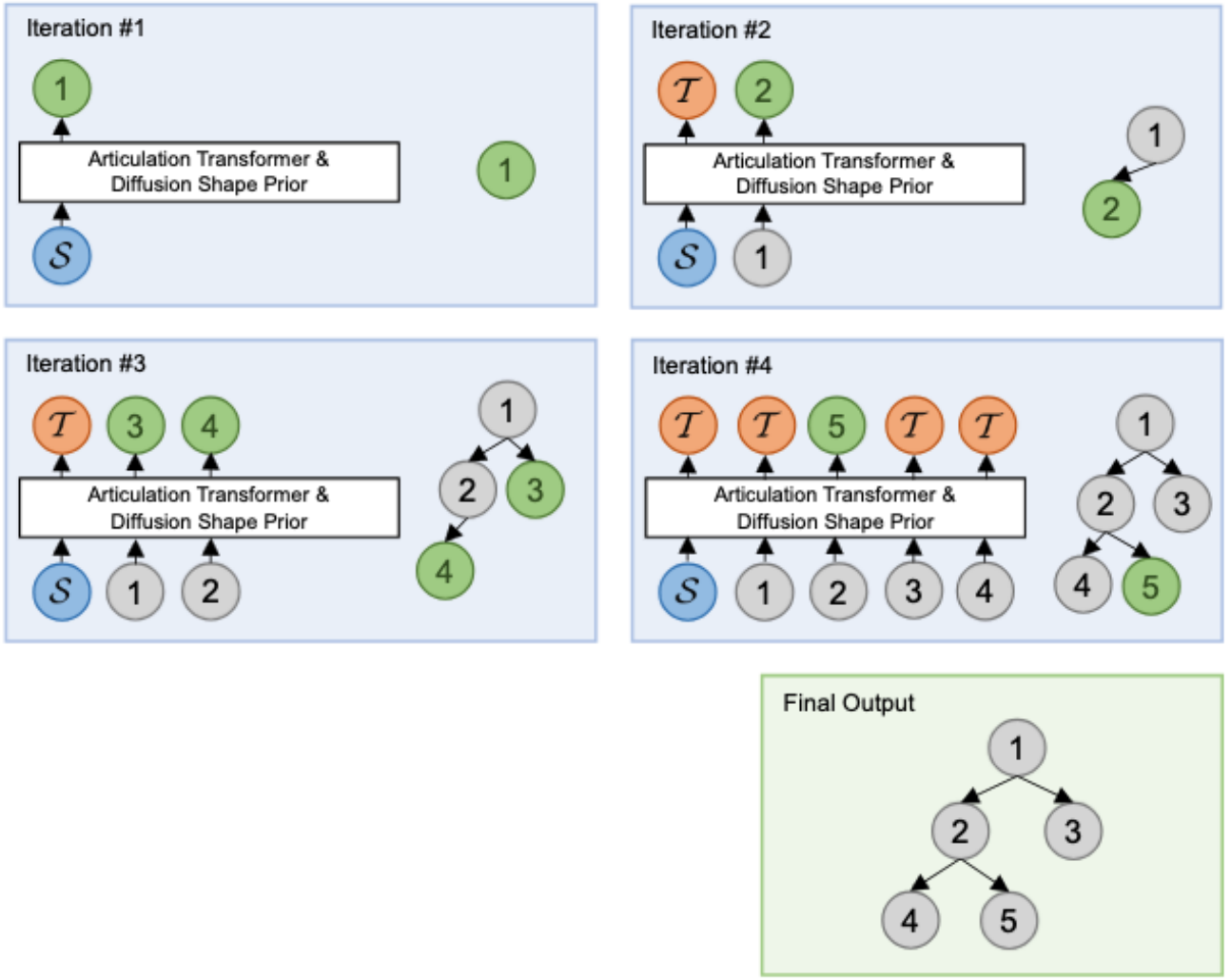
Idea: Sample Tree of Data with Transformers



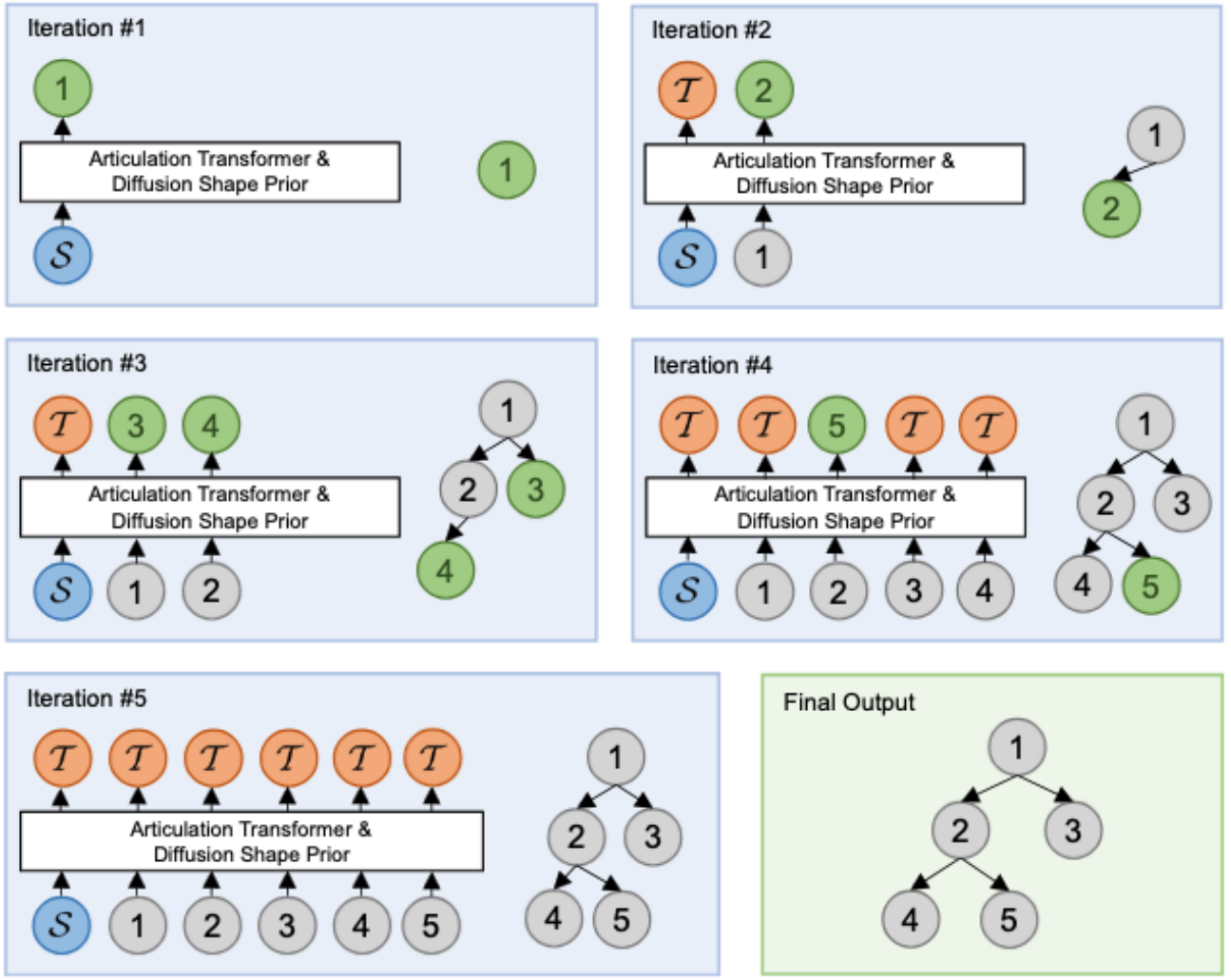


ArtFormer: Controllable Generation of Diverse 3D Articulated Objects. Su et al.

Idea: Sample Tree of Data with Transformers

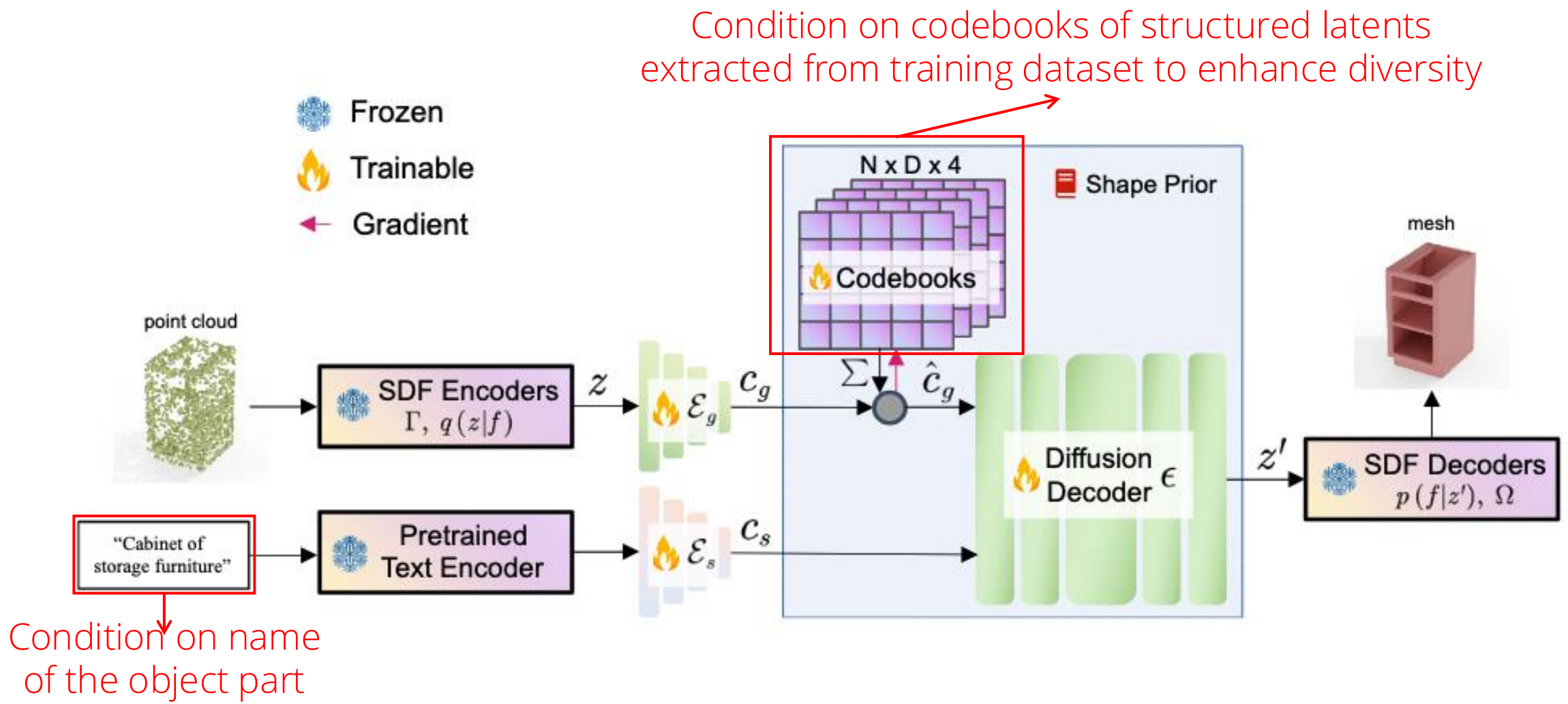


Idea: Sample Tree of Data with Transformers



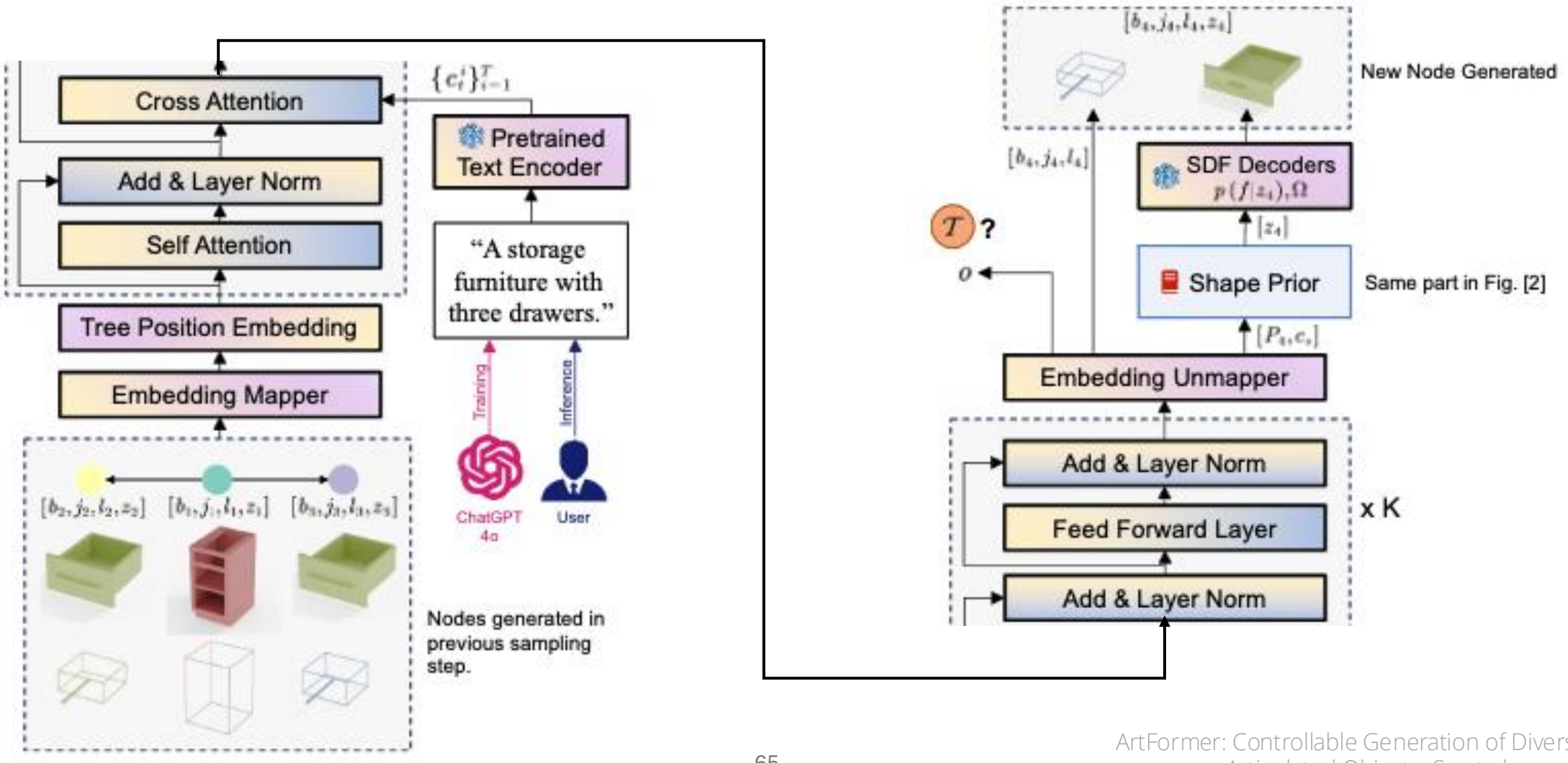
ArtFormer: Controllable Generation of Diverse 3D Articulated Objects. Su et al.

Learn Structured Latents with VAE

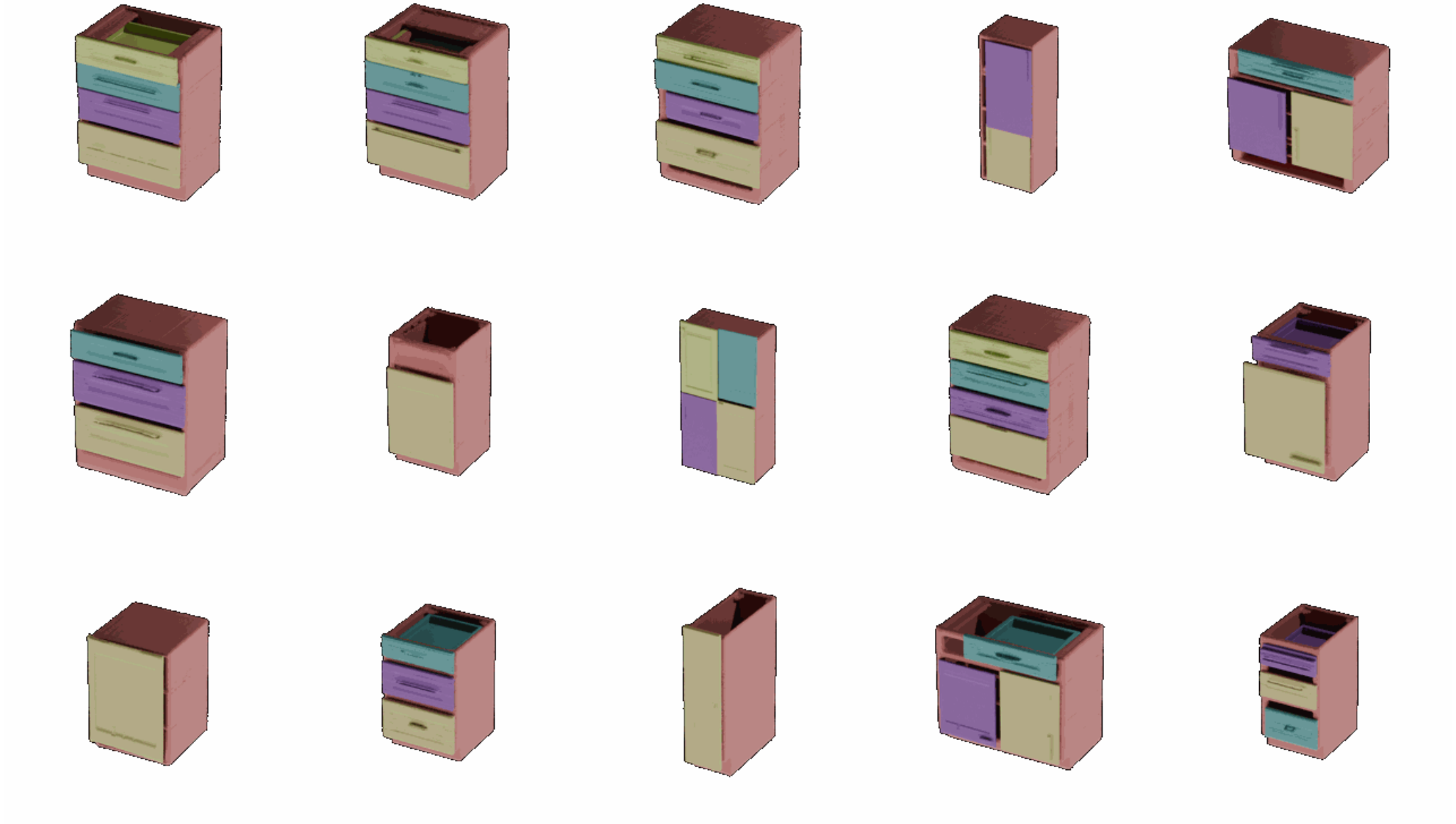


ArtFormer: Controllable Generation of Diverse 3D Articulated Objects. Su et al.

Put Everything Together



ArtFormer: Controllable Generation of Diverse 3D Articulated Objects. Su et al.



ArtFormer: Controllable Generation of Diverse 3D Articulated Objects. Su et al.

A Harder Problem: Generate 3D Articulated Objects from Scratch

We need to generate:

1. The connectivity graph of parts
2. The geometry and appearance of parts
3. The kinematics of connection (positions, directions and types of joints)

Image source: FreeArt3D: Training-Free Articulated Object Generation using 3D Diffusion. Chen et al.

Resources

- You can find additional information from
  - Physics Simulation in Visual Computing (with Javascript tutorial)
  - Physics-Based Animation Site by Cristopher's Batty

What We Will Cover the Next Week

- Physical modeling
  - Time Integration
  - Mass-Spring System
  - Position-Based Dynamics
  - The Finite Element Method