Represent, Reconstruct and Generate the 4D Real World

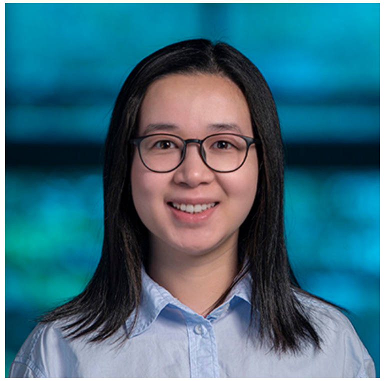
Jiahui Lei

2024 Sep

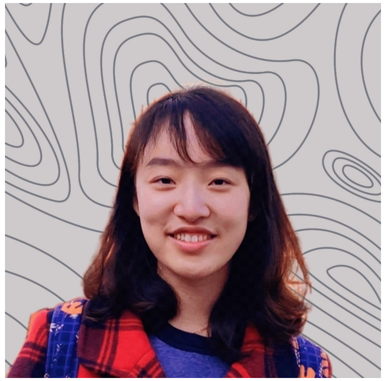
Main Contributors for today's work presented

Kostas Daniilidis
UPenn

Leonidas Guibas
Stanford



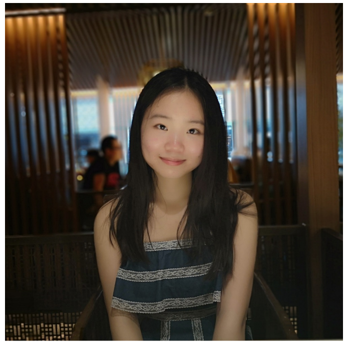
Lingjie Liu
UPenn



Congyue Deng
Stanford

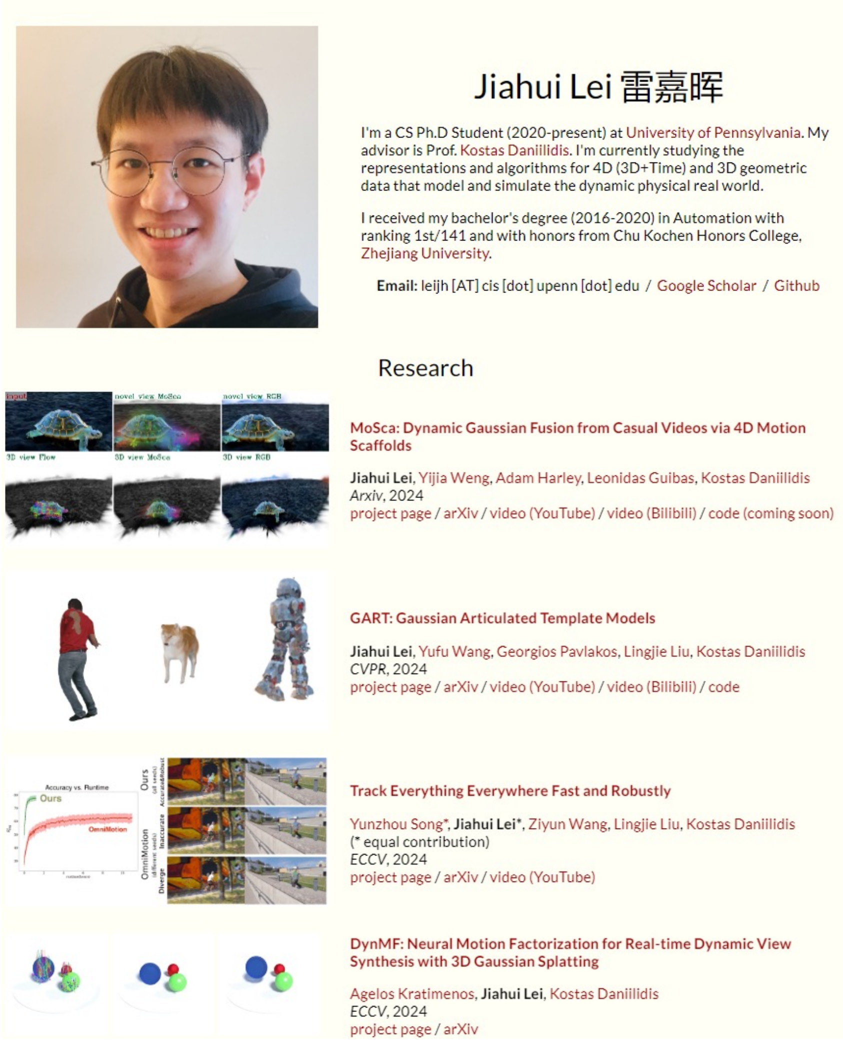
Yufu Wang
UPenn

Agelos Kratimenos
UPenn



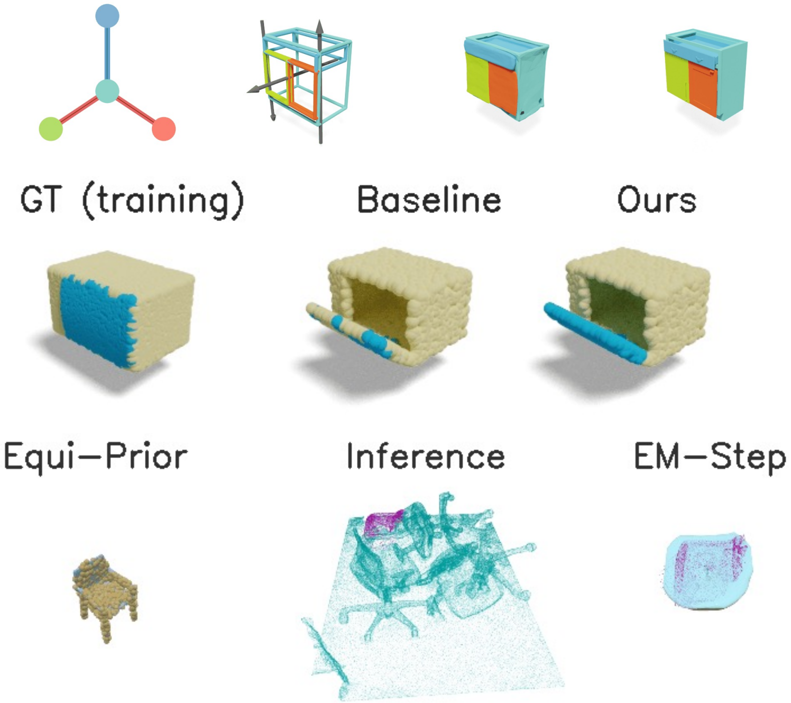
Yijia Weng
Stanford

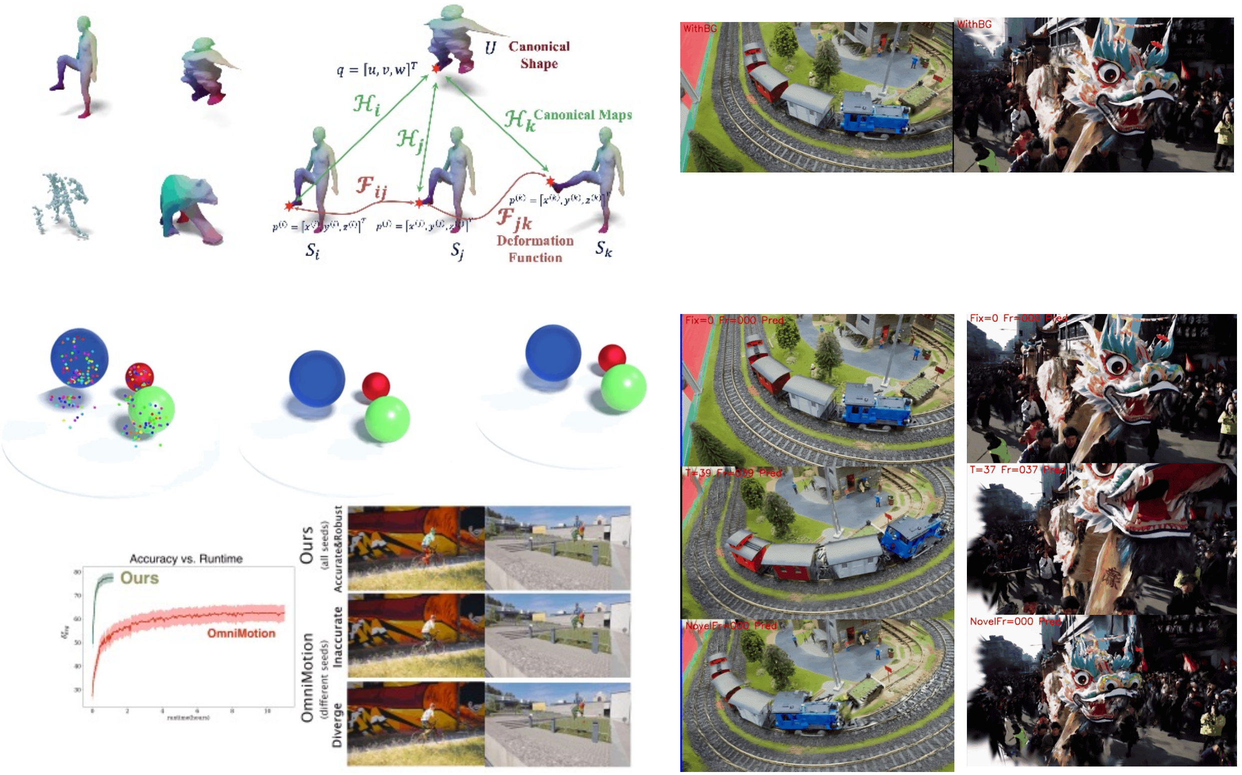
Yunzhou Song
UPenn



[Jiahui Lei 雷嘉晖 (upenn.edu)](https://www.cis.upenn.edu/~leijh/)

4D Modeling with Structure





MultiBody & Articulated Objects and Scenes

Semi-Articulated Objects

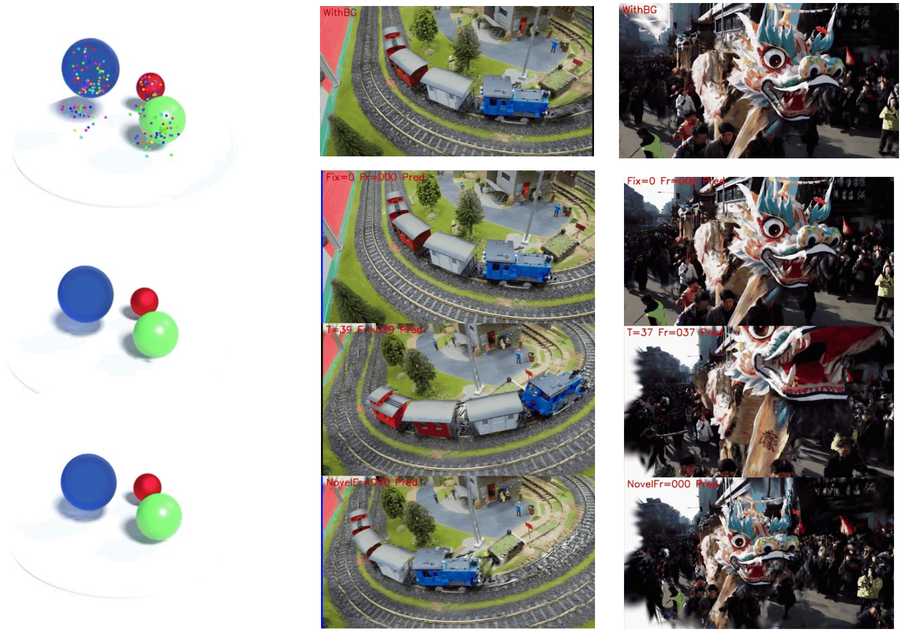
General Non-Rigid Object and Scenes

The next big step of the 3D vision community is 4D – the dynamic real world perception.
But dynamic vision/graphics problems are usually high dimensional – We need the "**Structure**"

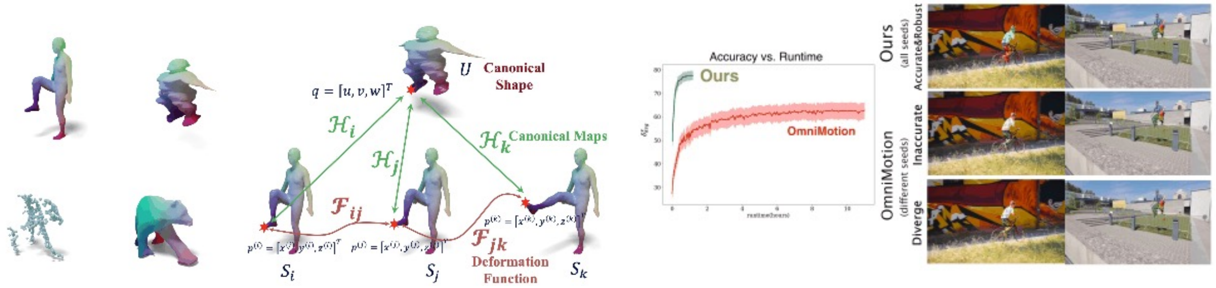
Overview of today’s talk

NAP: Neural Articulation Prior

GART: Gaussian Articulated Template Models



DynMF and MoSca

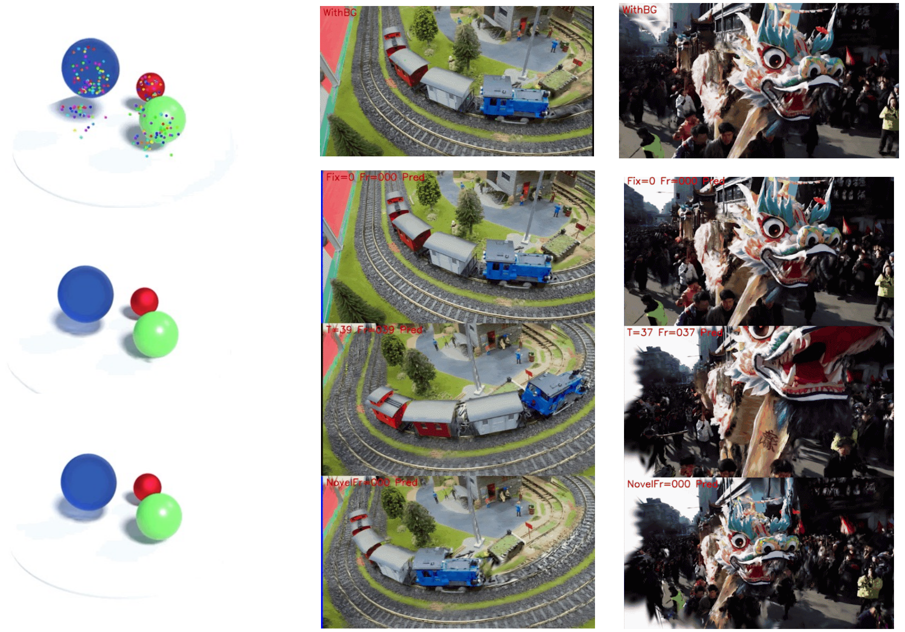


CaDeX and CaDeX++

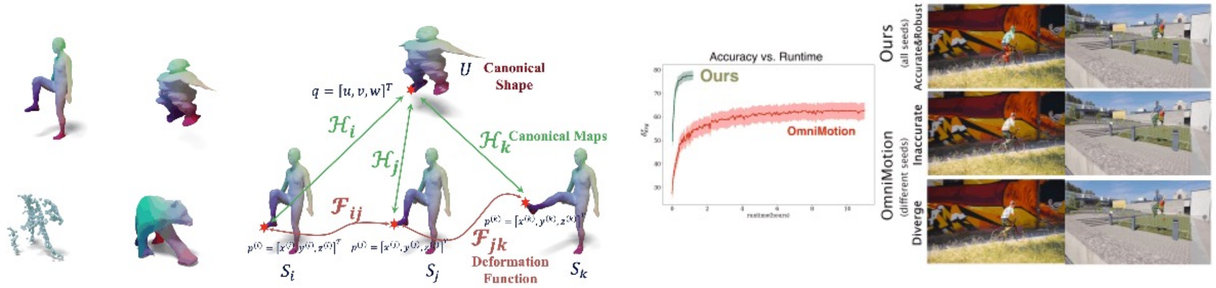
Overview of today’s talk

NAP: Neural Articulation Prior

GART: Gaussian Articulated Template Models



DynMF and MoSca

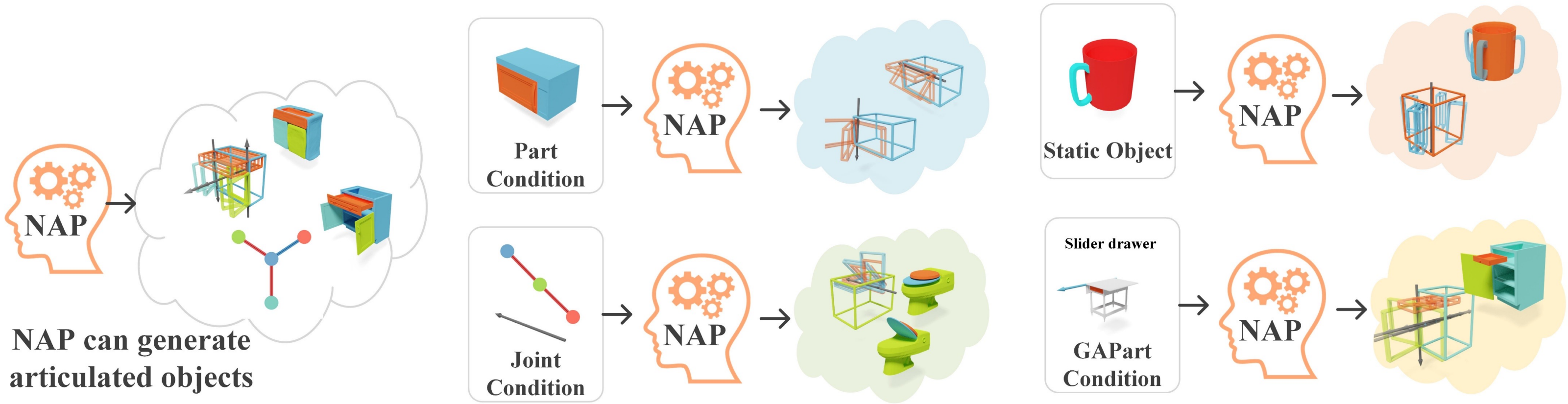


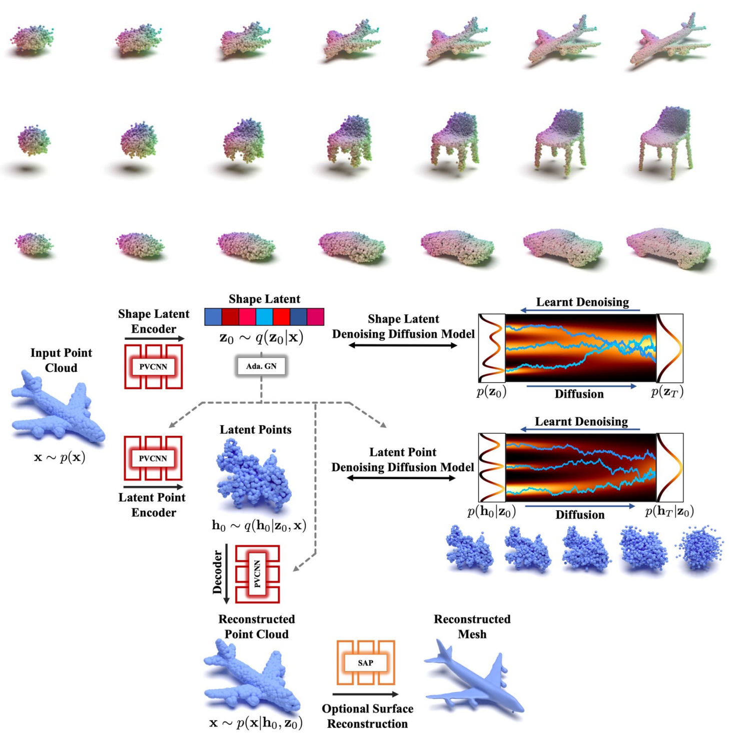
CaDeX and CaDeX++

NAP: Neural 3D Articulation Prior

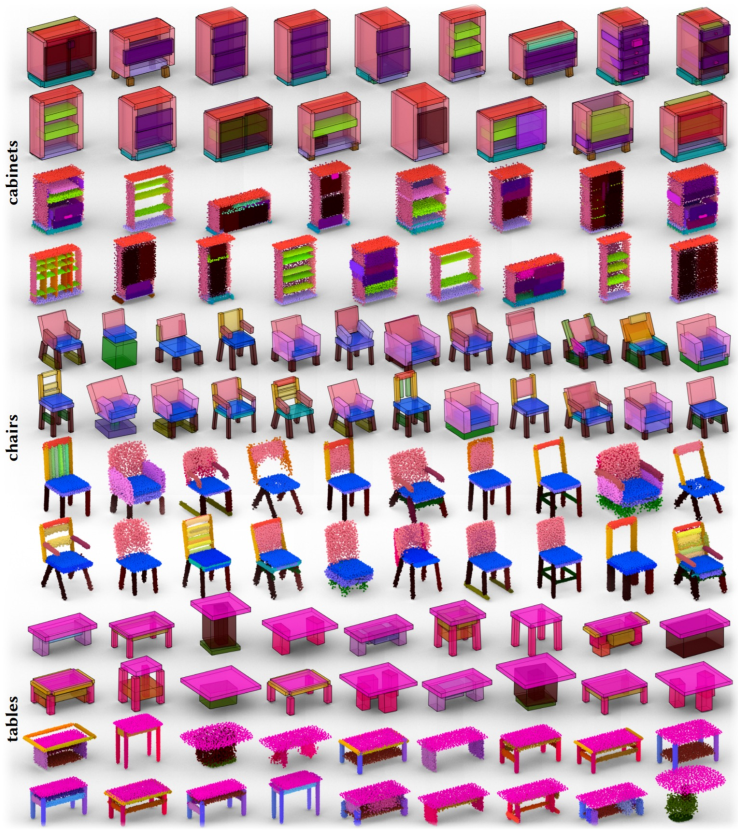
Jiahui Lei Congyue Deng Bokui Shen Leonidas Guibas Kostas Daniilidis

Quick Intro + Results **with narration**

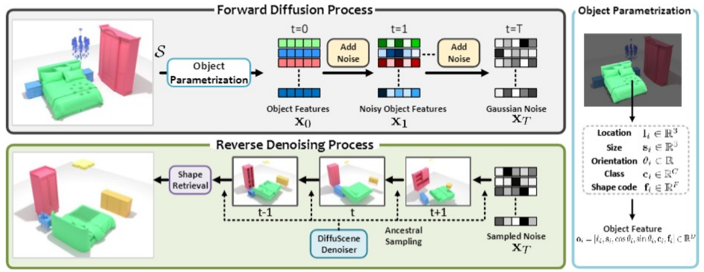




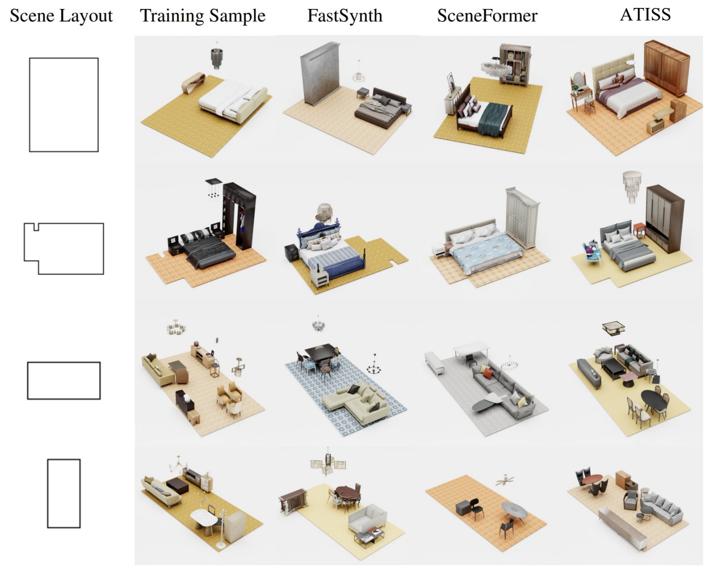
Shape Latent
Shape Latent Encoder
PVCNN
Input Point Cloud
$\mathbf{x} \sim p(\mathbf{x})$
$\mathbf{z}_0 \sim q(\mathbf{z}_0|\mathbf{x})$
Ada. GN
Shape Latent Denoising Diffusion Model
Learnt Denoising
Diffusion
$p(\mathbf{z}_0)$
$p(\mathbf{z}_T)$
Latent Point Encoder
Latent Points
$\mathbf{h}_0 \sim q(\mathbf{h}_0|\mathbf{z}_0, \mathbf{x})$
Latent Point Denoising Diffusion Model
$p(\mathbf{h}_0|\mathbf{z}_0)$
$p(\mathbf{h}_T|\mathbf{z}_0)$
Decoder
Reconstructed Point Cloud
$\mathbf{x} \sim p(\mathbf{x}|\mathbf{h}_0, \mathbf{z}_0)$
SAP
Optional Surface Reconstruction
Reconstructed Mesh



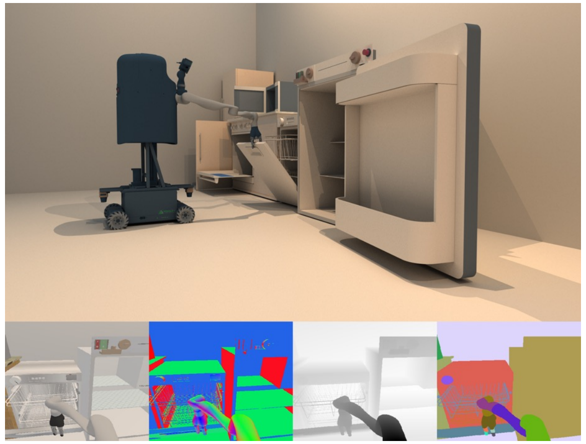
cabinets
chairs
tables

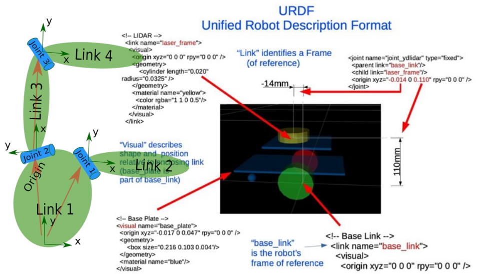


Forward Diffusion Process
Object Parametrization
Add Noise
Object Features
Noisy Object Features
Gaussian Noise
Reverse Denoising Process
Shape Retrieval
DiffuScene Denoiser
Ancestral Sampling
Sampled Noise
Location
Size
Orientation
Class
Shape code
Object Feature



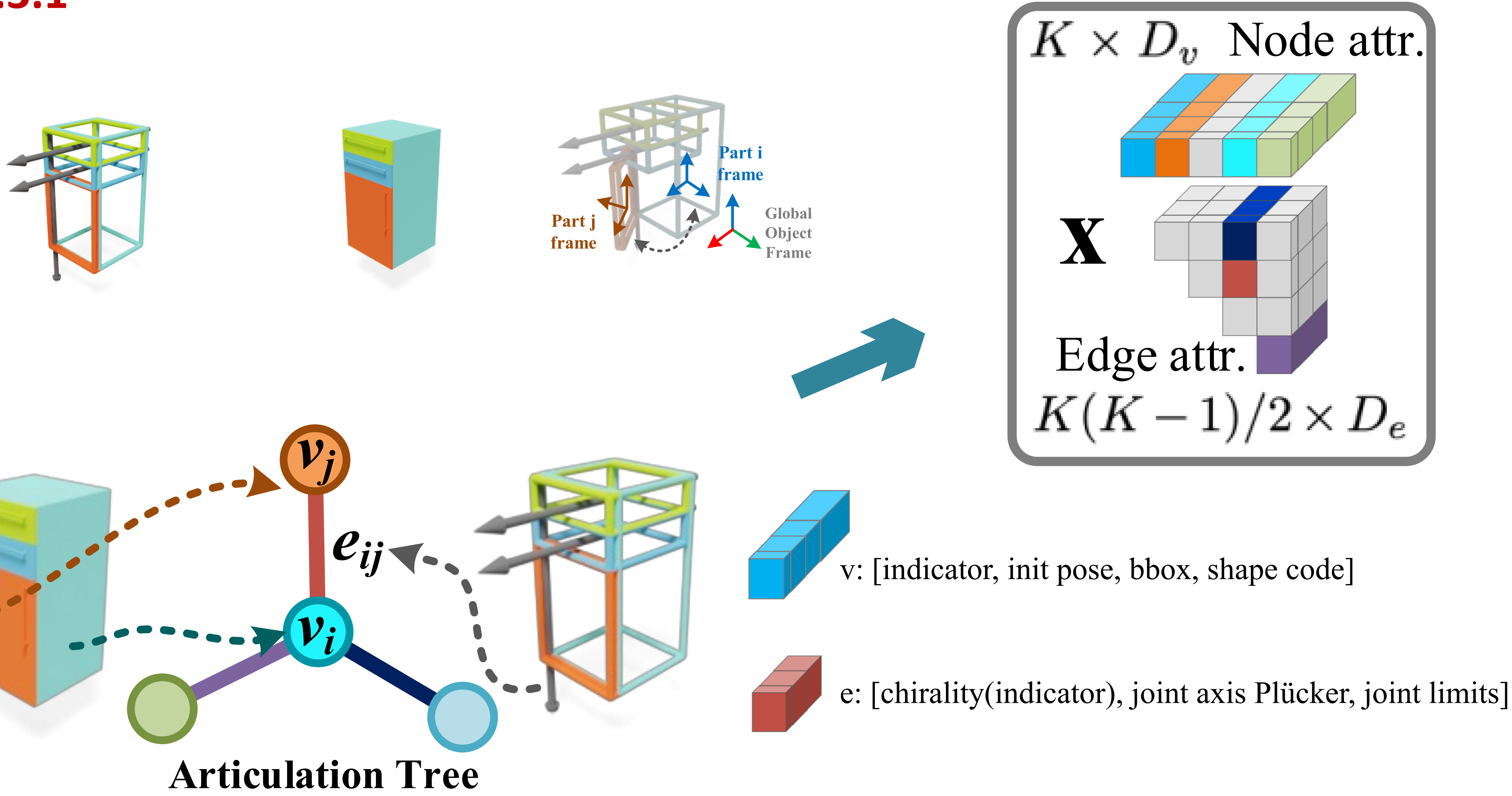
Scene Layout
Training Sample
FastSynth
SceneFormer
ATISS



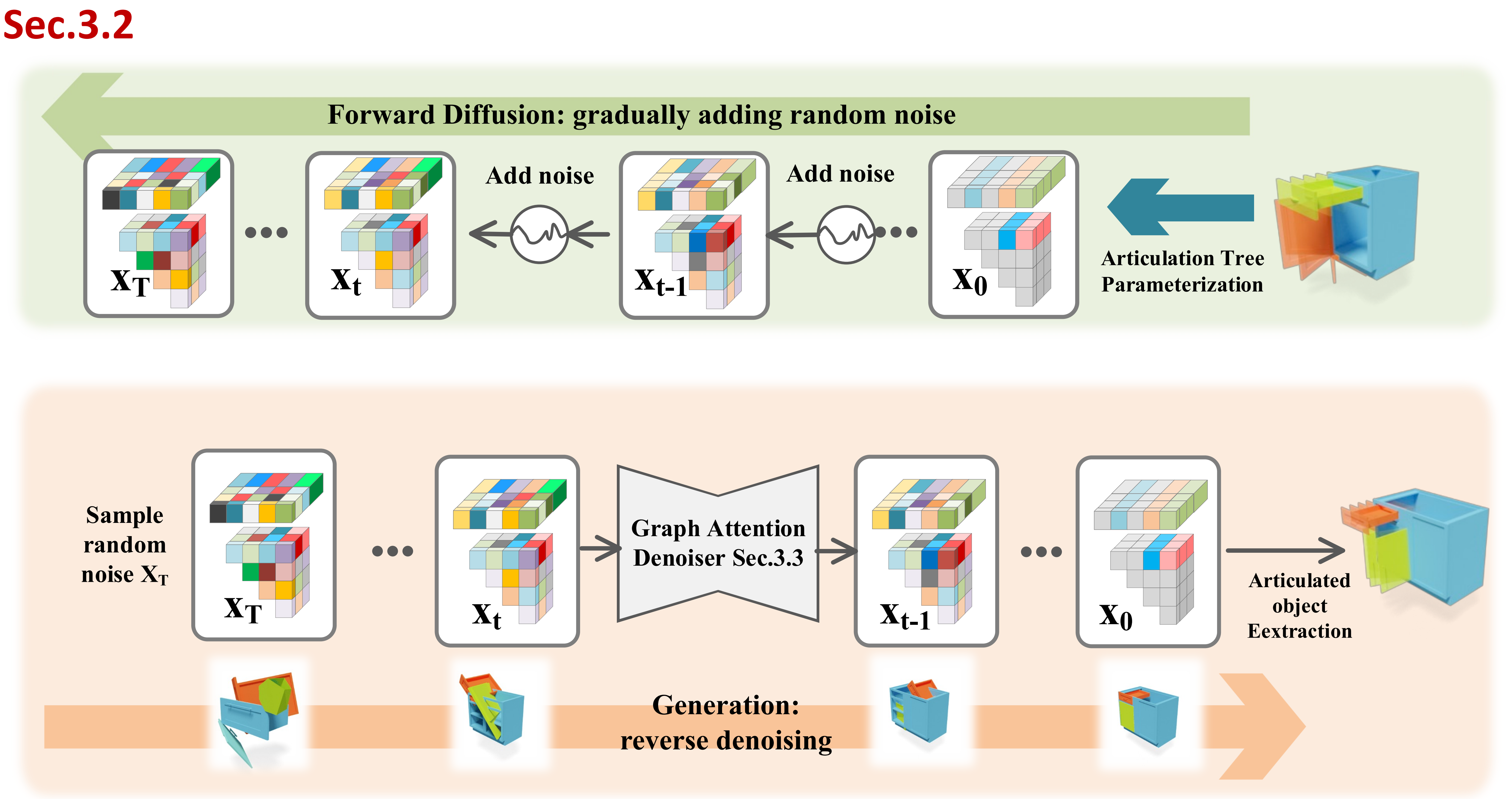


URDF
Unified Robot Description Format
Link 4
Link 3
Link 2
Link 1
Origin
"Link" identifies a Frame (of reference)
"Visual" describes shape and position
"base_link" is the robot's frame of reference
-14mm
110mm

Sec.3.1

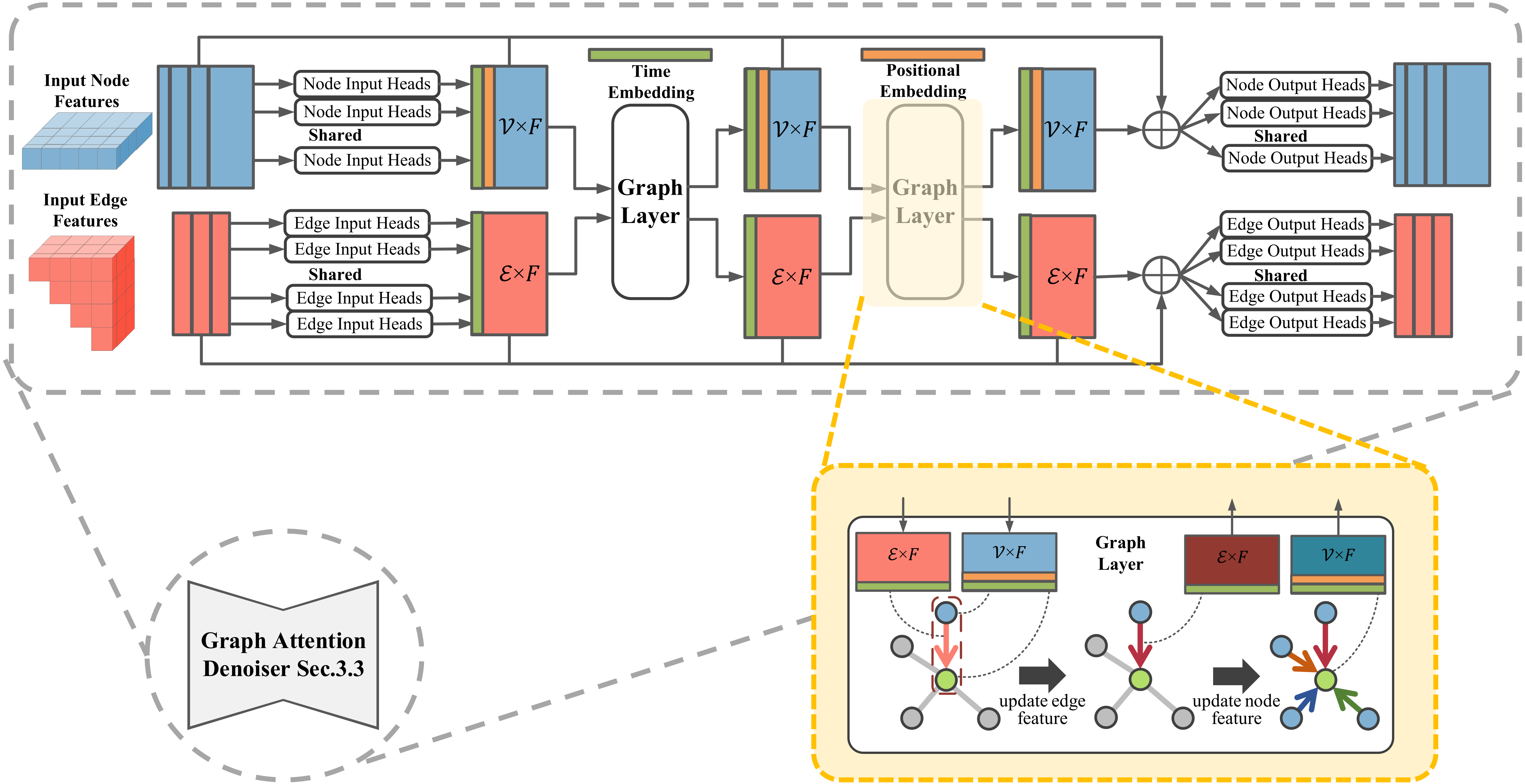


Part i frame
Part j frame
Global Object Frame
$K \times D_v$ Node attr.
X
Edge attr.
$K(K-1)/2 \times D_e$
v_j
e_{ij}
v_i
Articulation Tree
v: [indicator, init pose, bbox, shape code]
e: [chirality(indicator), joint axis Plücker, joint limits]



Sec.3.2
Forward Diffusion: gradually adding random noise
Add noise
Add noise
x_T
x_t
x_{t-1}
x_0
Articulation Tree Parameterization
Sample random noise X_T
Graph Attention Denoiser Sec.3.3
Articulated object Eextraction
Generation: reverse denoising

Sec.3.3


Input Node Features
Input Edge Features
Node Input Heads
Node Input Heads
Shared
Node Input Heads
Edge Input Heads
Edge Input Heads
Shared
Edge Input Heads
Edge Input Heads
$\mathcal{V}\times F$
$\mathcal{E}\times F$
Time Embedding
Graph Layer
Positional Embedding
Graph Layer
Node Output Heads
Node Output Heads
Shared
Node Output Heads
Edge Output Heads
Edge Output Heads
Shared
Edge Output Heads
Edge Output Heads
Graph Attention Denoiser Sec.3.3
Graph Layer
update edge feature
update node feature

Sec.4.1

We treat an articulated object O as a template that, given the joint states $q \in \mathcal{Q}_O$ in object's joint range $\mathcal{Q}_O$, it returns the overall articulate mesh $\mathcal{M}(q)$ and the list of part poses $\mathcal{T}(q) = \{T_{\text{part}} \in SE(3)\}$. We compute the distance between two articulated objects in different joint states by

$$\tilde{d}(O_1, q_1, O_2, q_2) = \min_{T_i \in \mathcal{T}_1(q_1), T_j \in \mathcal{T}_2(q_2)} \left\{ D(T_i^{-1}\mathcal{M}_1(q_1), T_j^{-1}\mathcal{M}_2(q_2)) \right\}, \tag{9}$$

where $T_i^{-1}\mathcal{M}_1(q_1)$ means canonicalizing the mesh using its ith part pose, and D is a standard distance that measures the distance between two static meshes. Specifically, we sample $N = 2048$ points from two meshes and compute their Chamfer Distance. Intuitively, the above distance measures the minimum distance between two posed articulated objects by trying all possible canonicalization combinations. Then, we define the instantiation distance between O_1 and O_2 as:

$$\begin{aligned} ID(O_1, O_2) =& \mathbb{E}_{q_1 \in \mathcal{U}(\mathcal{Q}_{O_1})} \left[\inf_{q_2 \in \mathcal{Q}_{O_2}} \left(\tilde{d}(O_1, q_1, O_2, q_2) \right) \right] \\ &+ \mathbb{E}_{q_2 \in \mathcal{U}(\mathcal{Q}_{O_2})} \left[\inf_{q_1 \in \mathcal{Q}_{O_1}} \left(\tilde{d}(O_1, q_1, O_2, q_2) \right) \right], \end{aligned} \tag{10}$$

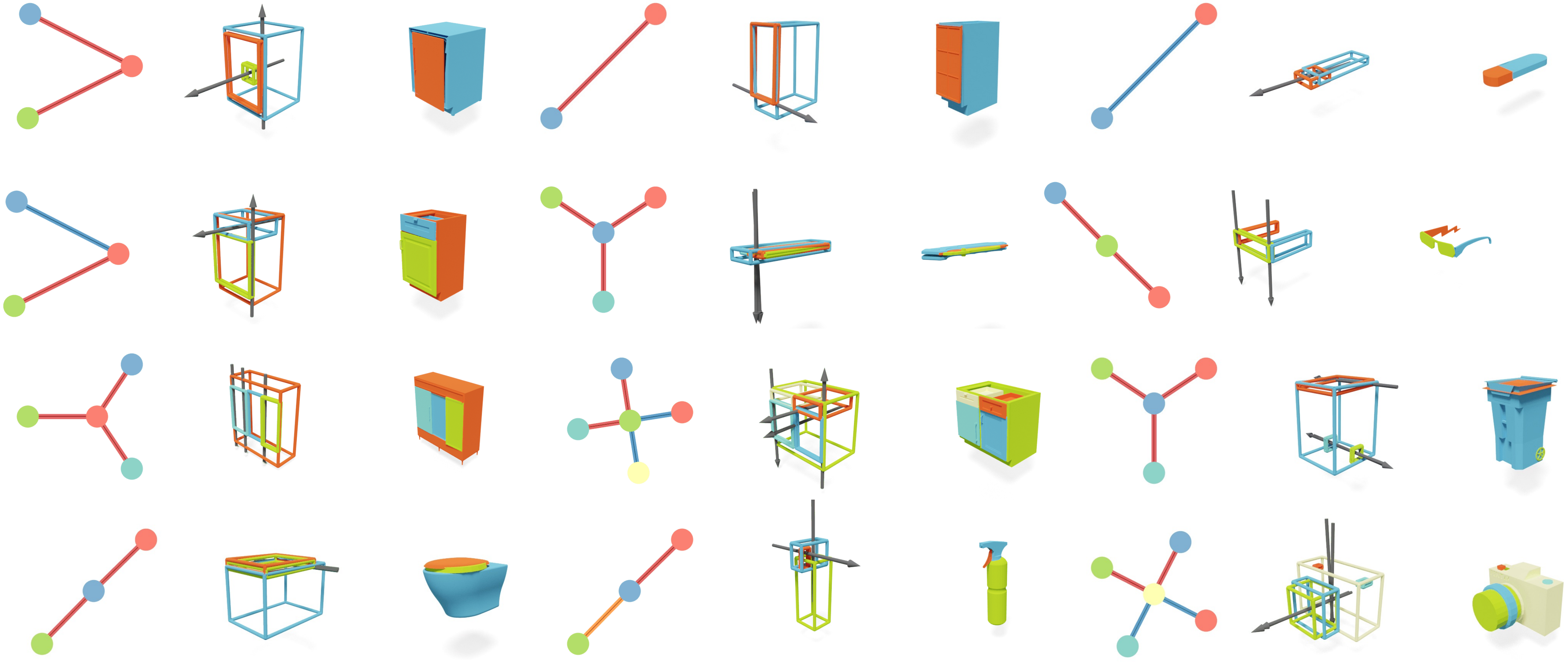
where $q \in \mathcal{U}(\mathcal{Q}_O)$ means uniformly sample joint poses from the joint states range. The instantiation

	Part SDF Shape			Part Retrieval Shape		
Generative Paradigm/Method	MMD ↓	COV ↑	1-NNA ↓	MMD ↓	COV ↑	1-NNA ↓
Auto-Decoding (StructNet)	0.0435	0.1871	0.8820	0.0390	0.2316	0.8675
Variational Auto-Encoding (StructNet)	0.0311	0.3497	0.8085	0.0289	0.3363	0.7918
Autoregressive (ATISS-Tree)	0.0397	0.3808	0.6860	0.0333	0.4120	0.6782
Latent Diffusion (StructNet)	0.0314	0.4365	0.6269	0.0288	0.4477	0.6102
Articulation Graph Diffusion (Ours)	**0.0268**	**0.4944**	**0.5690**	**0.0215**	**0.5234**	**0.5412**

Table 1: Articualted object synthesis comparison with Instantiation Distance

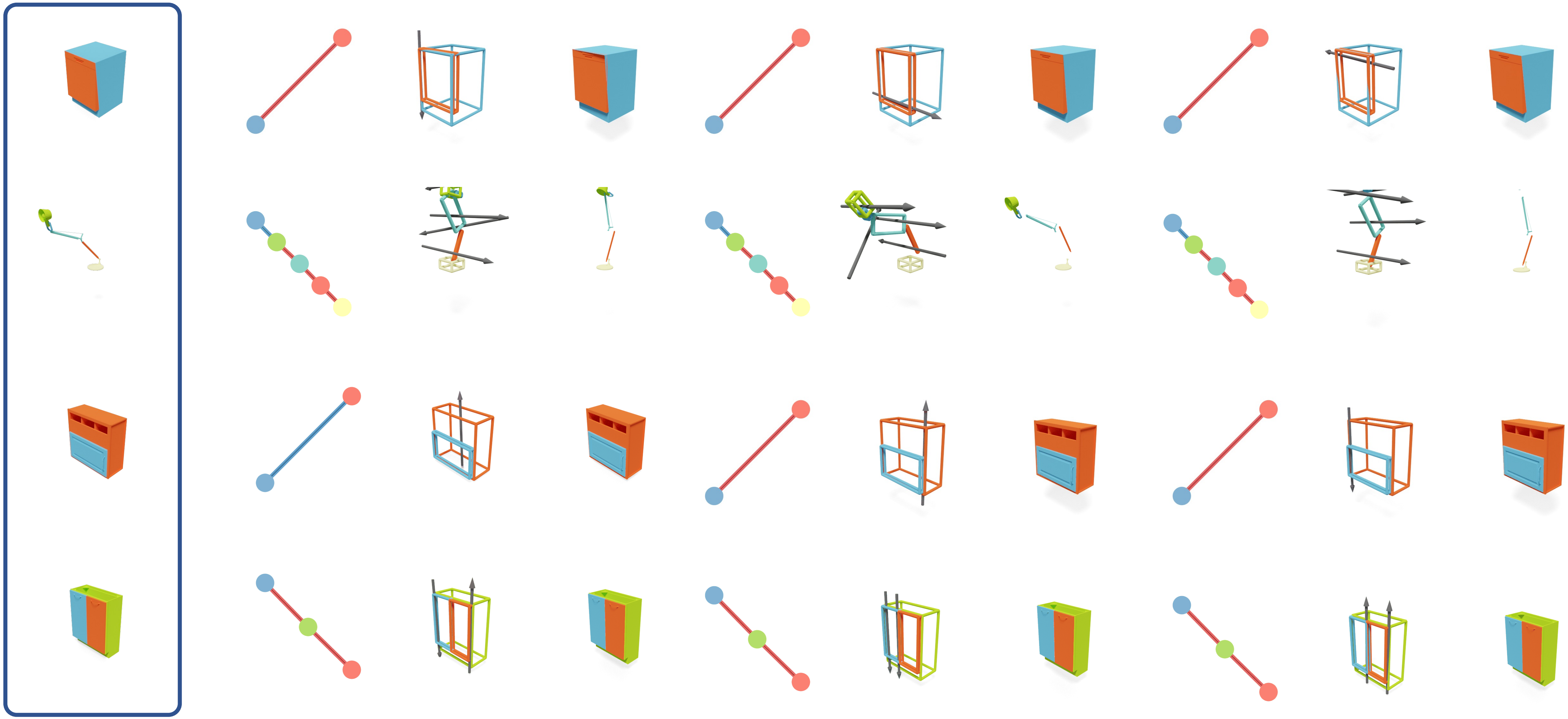
Sec.4.2

Articulated object synthesis



Sec.4.4

Part2Motion

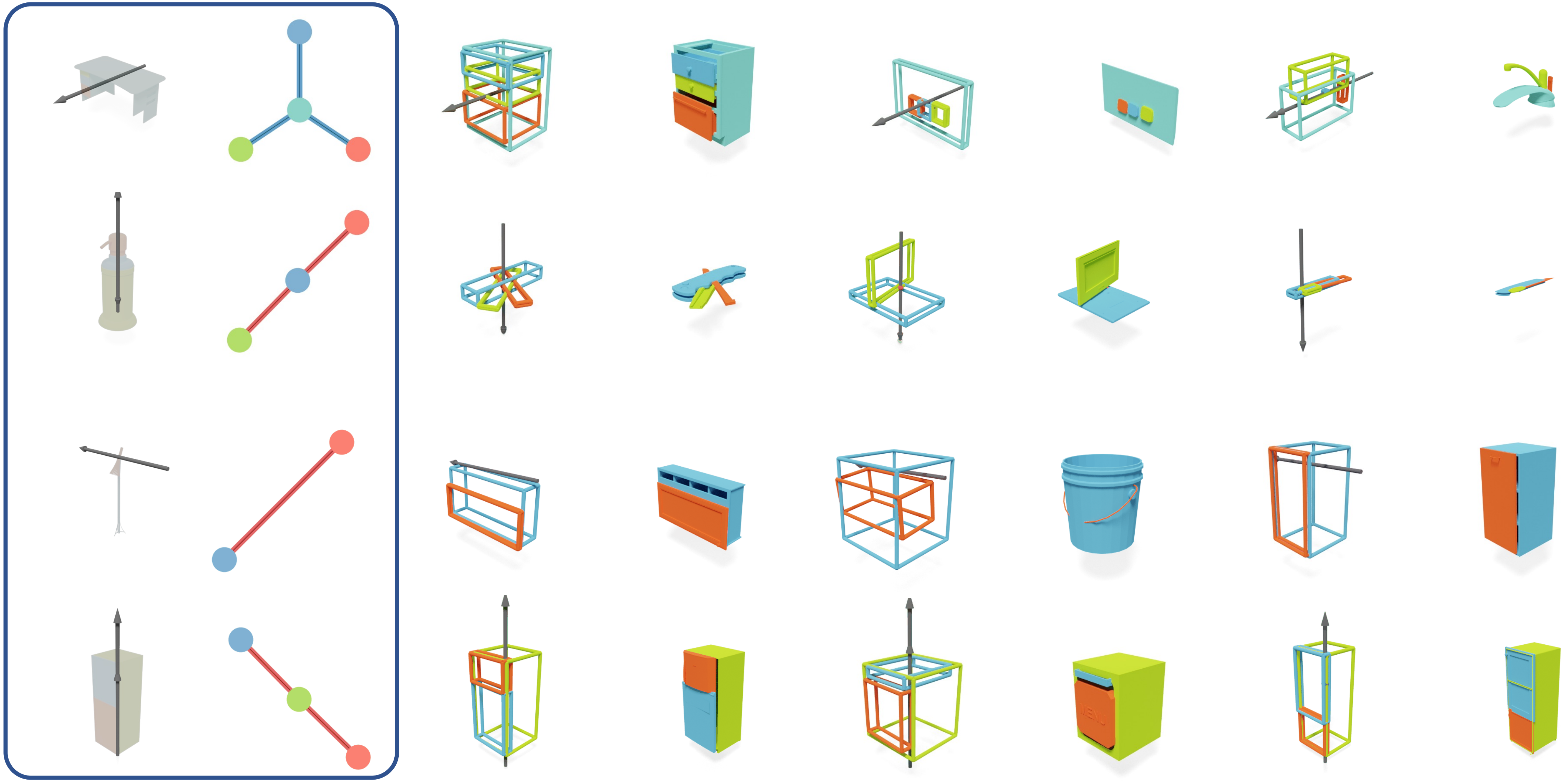


Sec.4.4

PartNet Imagination

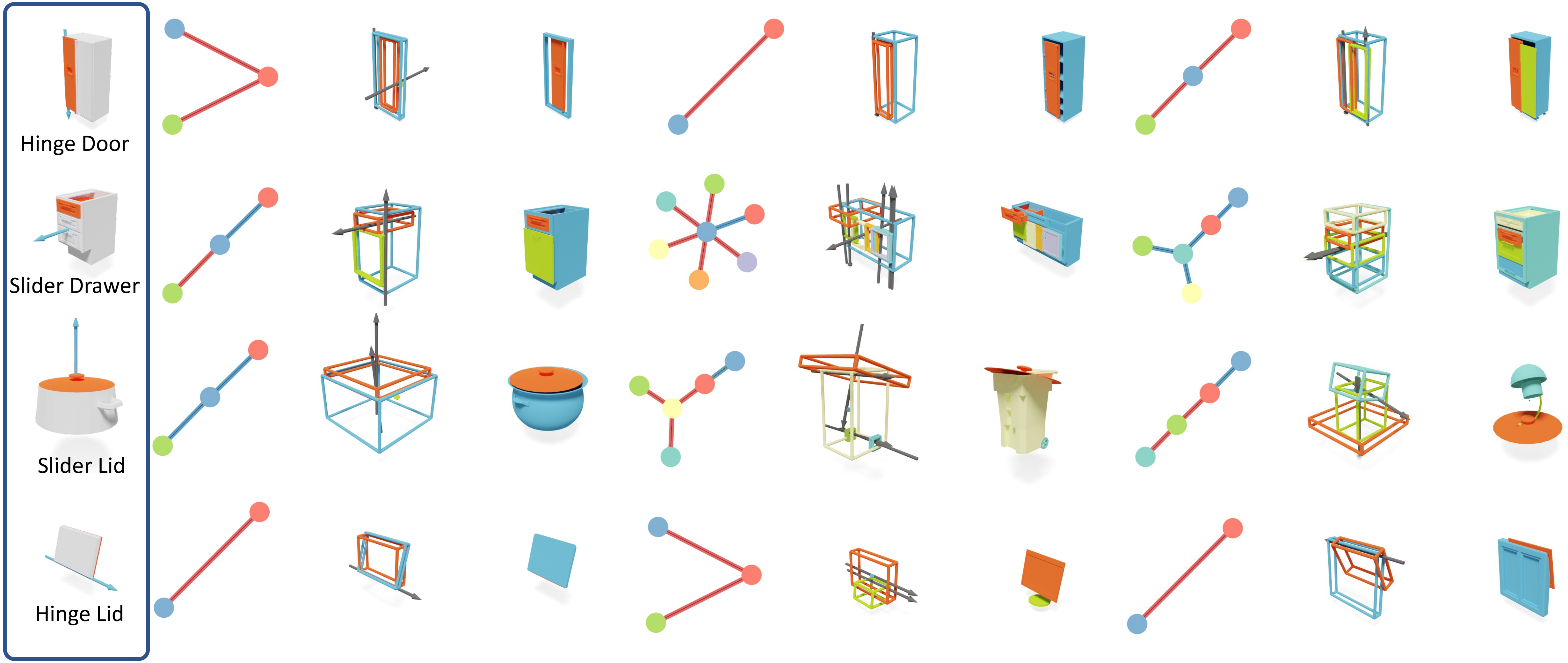
Sec.4.4

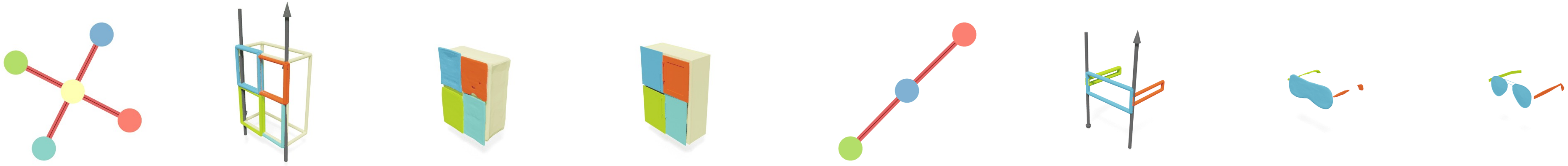
Motion2Part



Sec.4.4

GAPart2Object

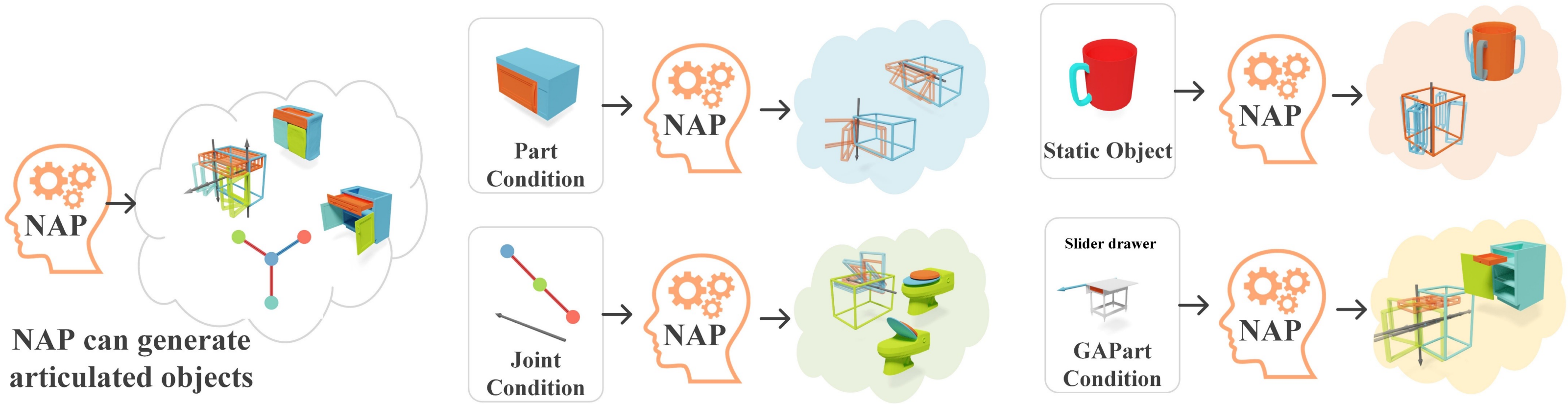




NAP: Neural 3D Articulation Prior

Jiahui Lei   Congyue Deng   Bokui Shen   Leonidas Guibas   Kostas Daniilidis

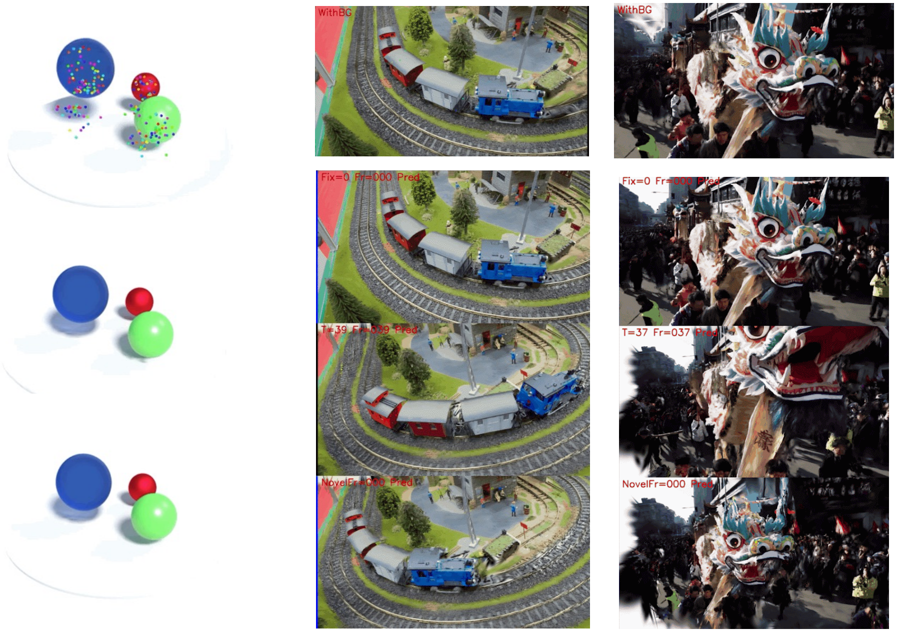
Quick Intro + Results **with narration**



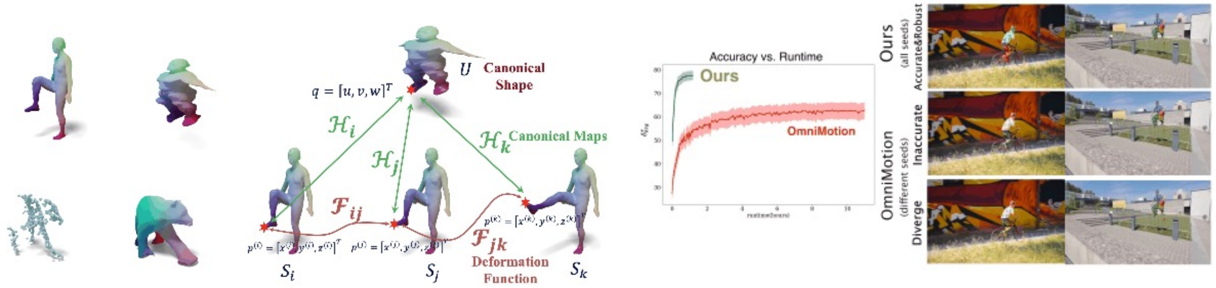
Overview of today’s talk

NAP: Neural Articulation Prior

GART: Gaussian Articulated Template Models



DynMF and MoSca



CaDeX and CaDeX++

The Rising of Point-Based Method

ZWICKER ET AL.: EWA SPLATTING

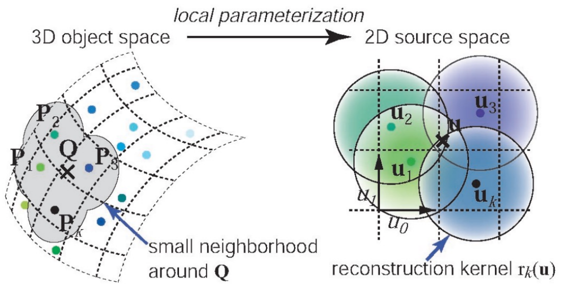


Fig. 6. Defining a texture function on the surface of a point-based object.

231

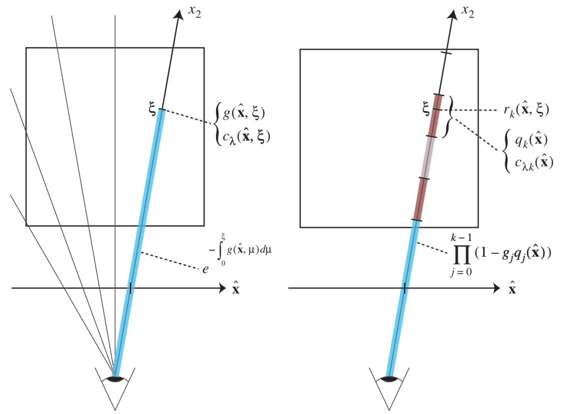


Figure 2: *Volume rendering. Left: Illustrating the volume rendering equation in 2D. Right: Approximations in typical splatting algorithms.*

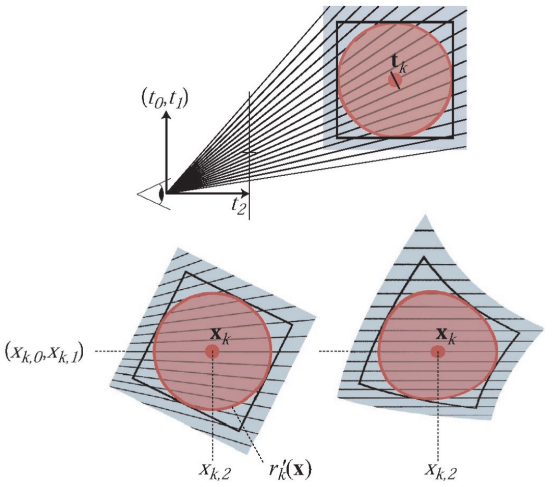
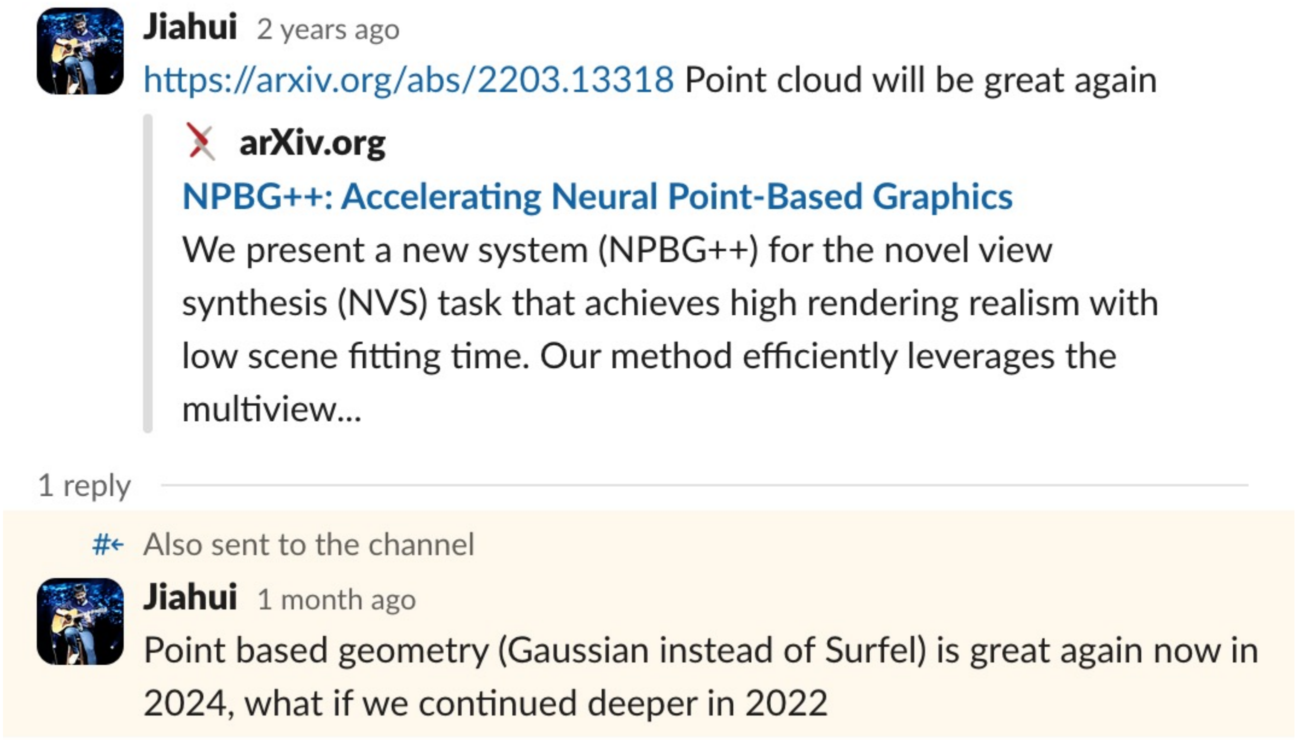


Fig. 9. Mapping a reconstruction kernel from camera to ray space. Top: camera space. Bottom: ray space. Left: local affine mapping. Right: exact mapping.



GART

*Ga*ussian *Art*iculated *T*emplate Models

Jiahui Lei **Yufu Wang** **Georgios Pavlakos**

Lingjie Liu **Kostas Daniilidis**


AUTO

Loper et al. SMPL 2015

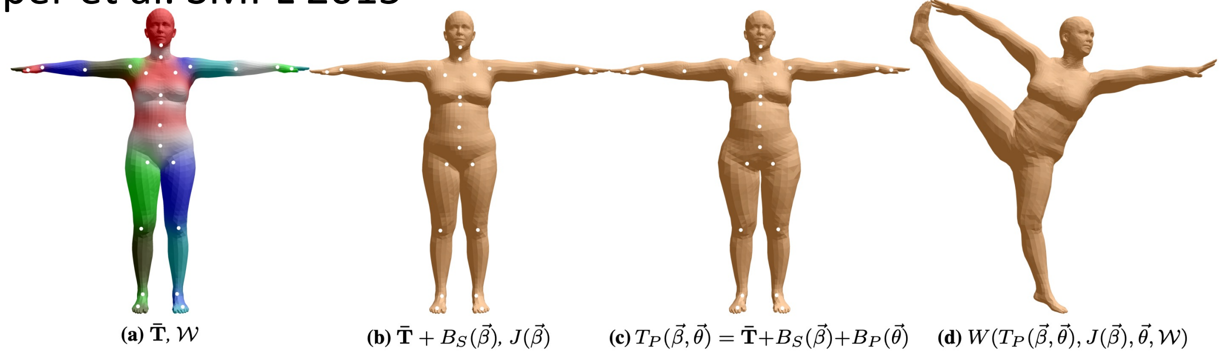


Figure 3: SMPL model. *(a) Template mesh with blend weights indicated by color and joints shown in white. (b) With identity-driven blendshape contribution only; vertex and joint locations are linear in shape vector $\vec{\beta}$. (c) With the addition of of pose blend shapes in preparation for the split pose; note the expansion of the hips. (d) Deformed vertices reposed by dual quaternion skinning for the split pose.*

Ruegg et al. BITE 2023

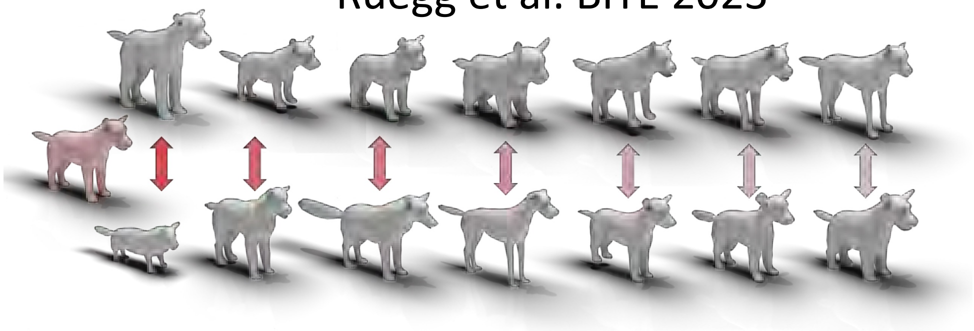
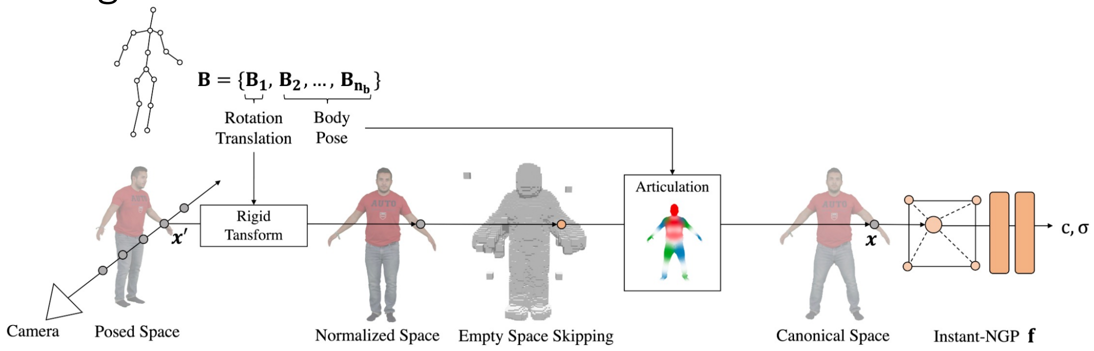
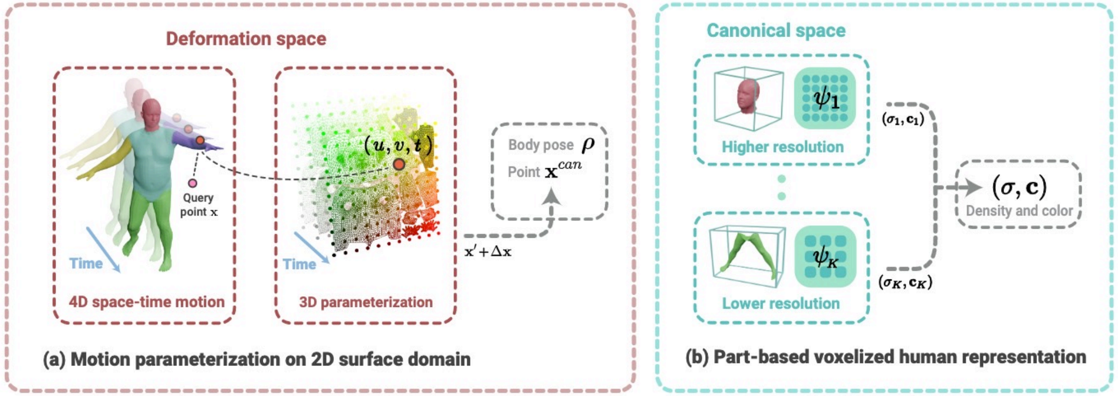


Figure 2. *D-SMAL shape space.* Shown are the mean shape and the 7 principal modes of deformation.

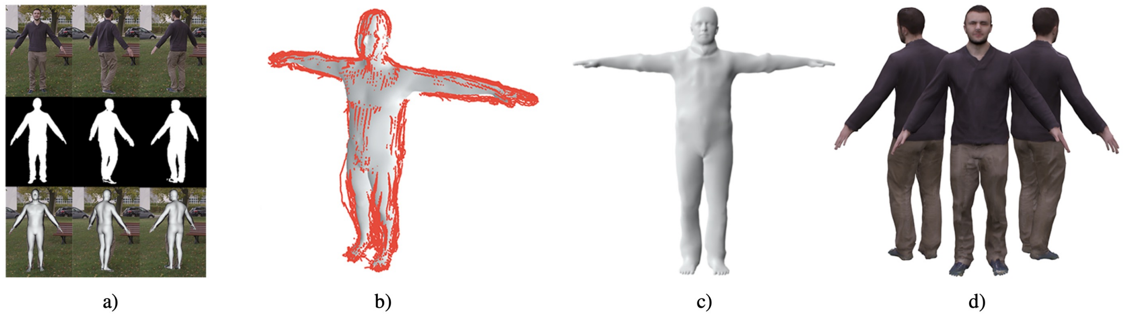
Jiang et al. InstantAvatar 2023

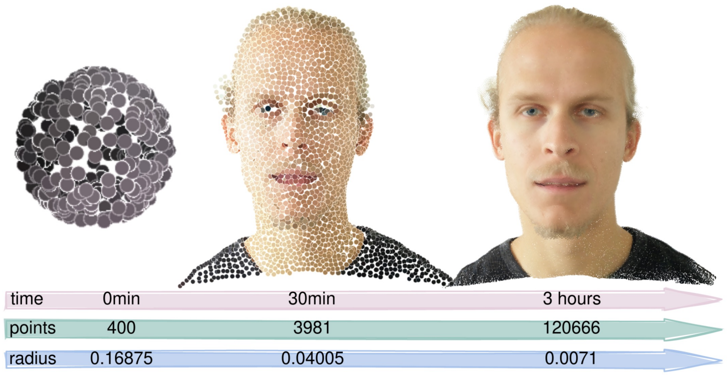




Geng et al. Instant-NVR 2023

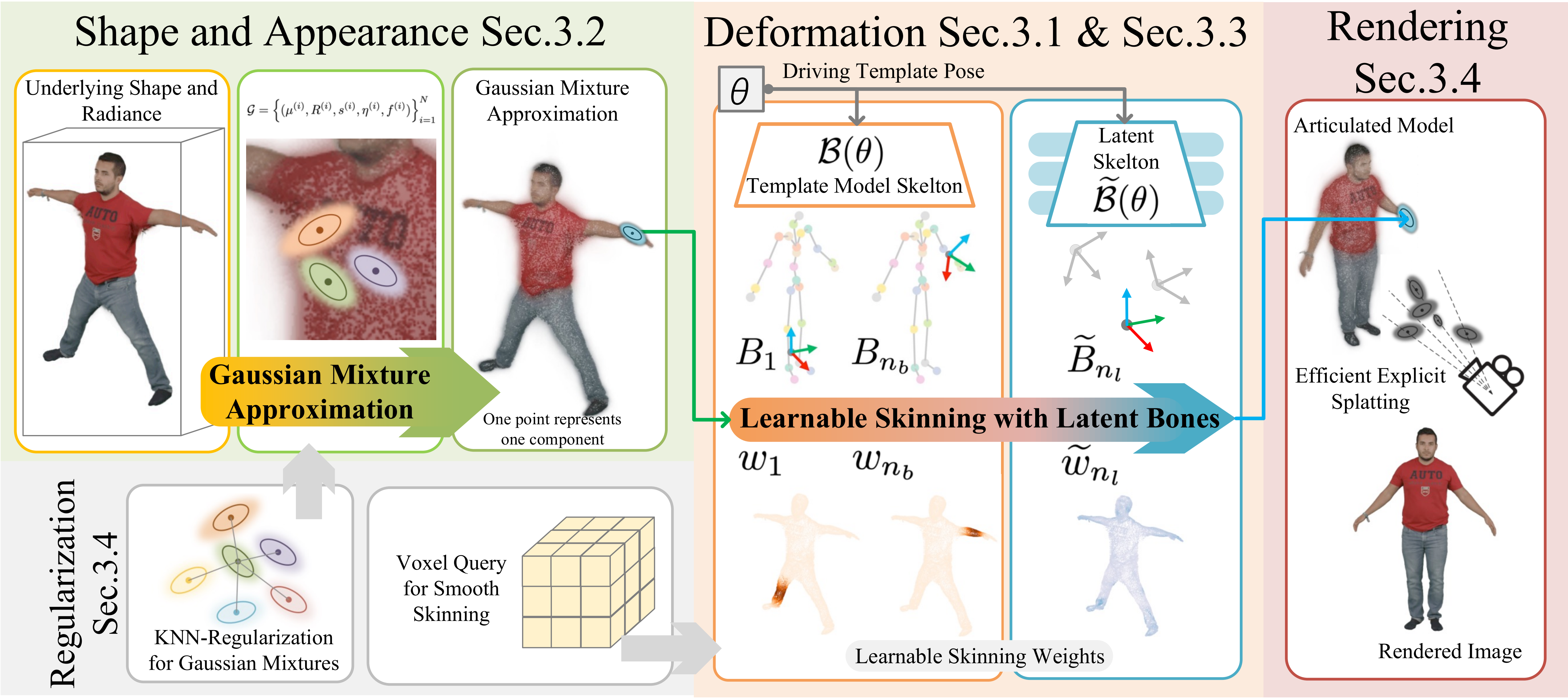
Alldieck et al. People Snapshot 2018



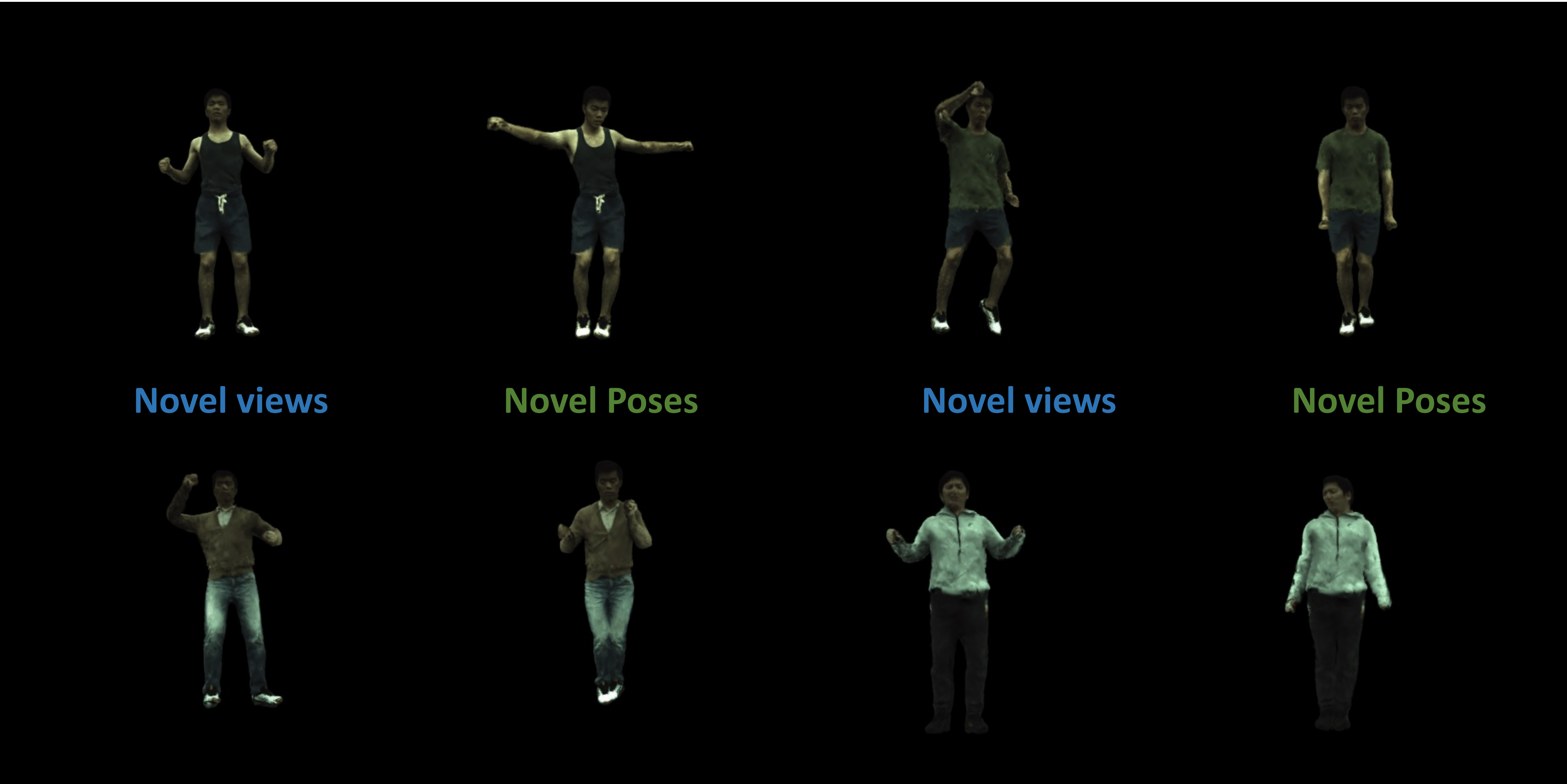


Zheng et al.
PointAvatar 2022

Method Overview



ZJU-MoCap Results



People-Snapshot Results [150+ Inference FPS]

Novel views

Novel Poses

Novel views

Novel Poses

More Challenging Mono-Sequences (UBC-Fashion)

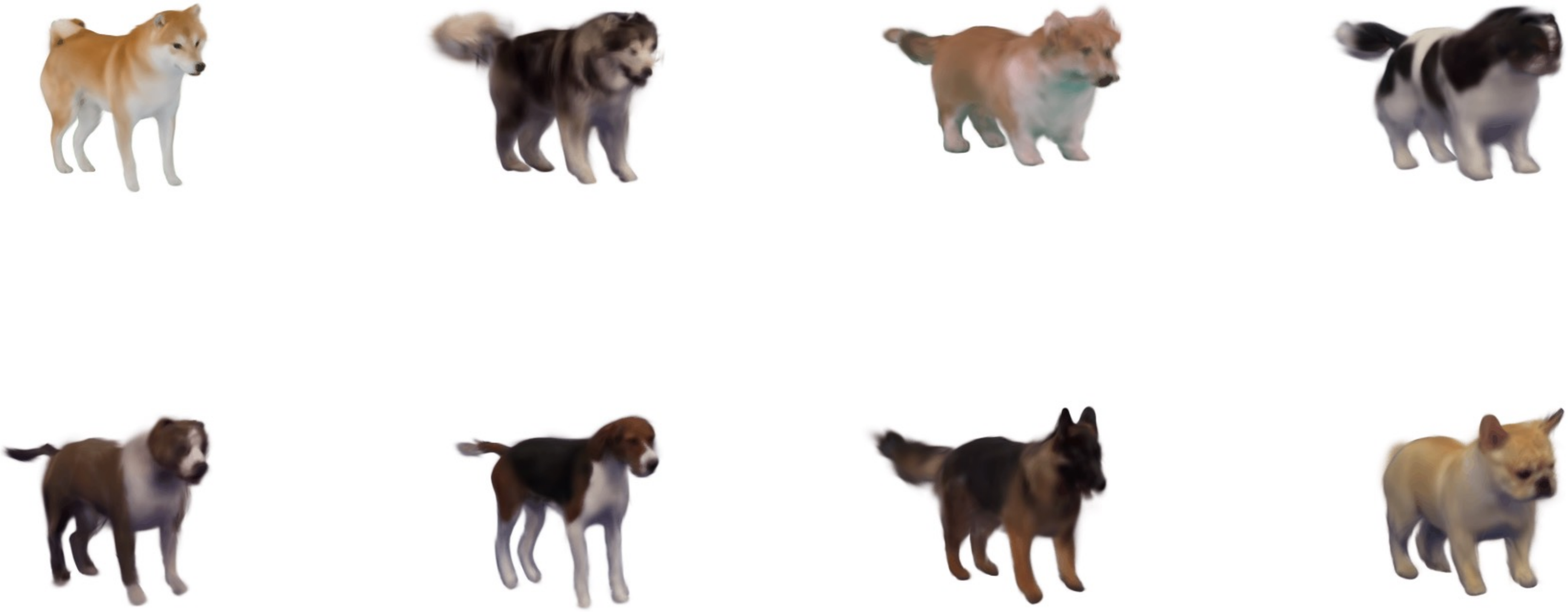
In-the-Wild Challenging Mono-Sequences

Input Video

Novel Views

Novel Poses

Diverse Dog Breeds



Application: Text-to-GART

A policeman in blue uniform

A doctor in green surgical uniform

Skywalker

A yellow CyberPunk robot, silver skeleton

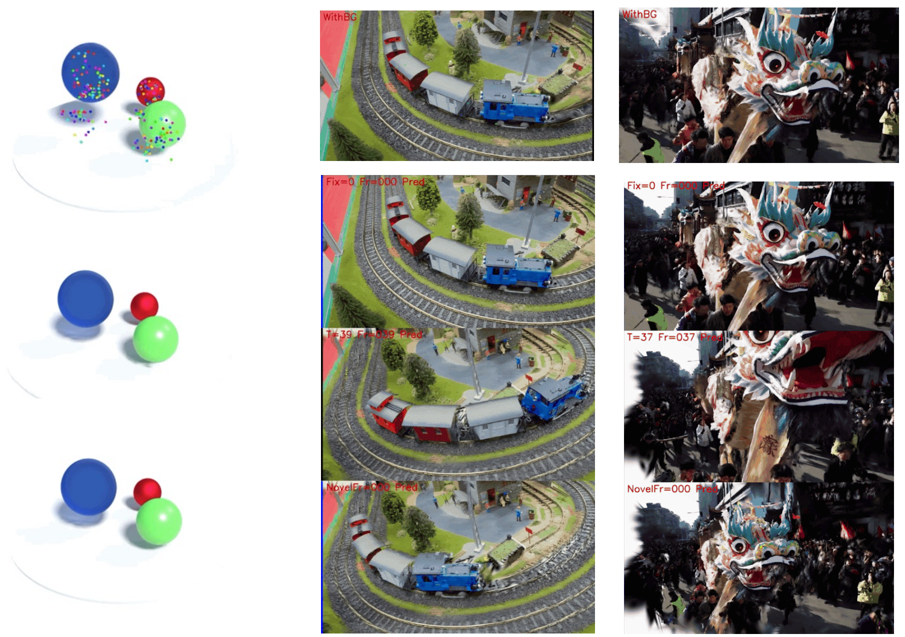
A frog character from a game

A silver robot with single red eye like hal9000

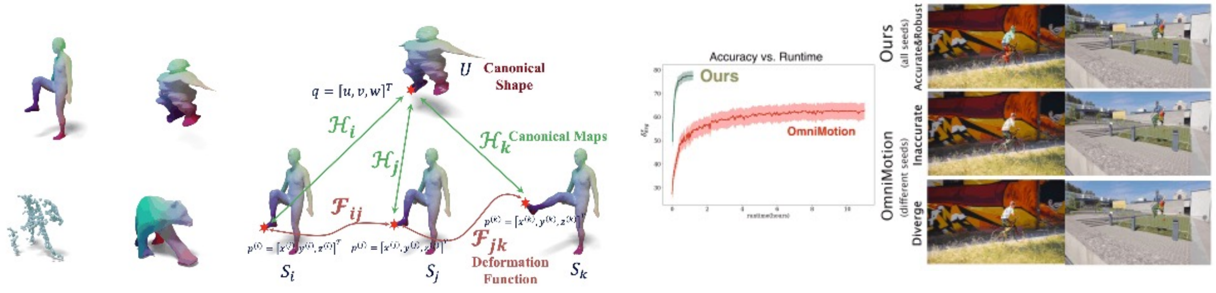
Overview of today’s talk

NAP: Neural Articulation Prior

GART: Gaussian Articulated Template Models



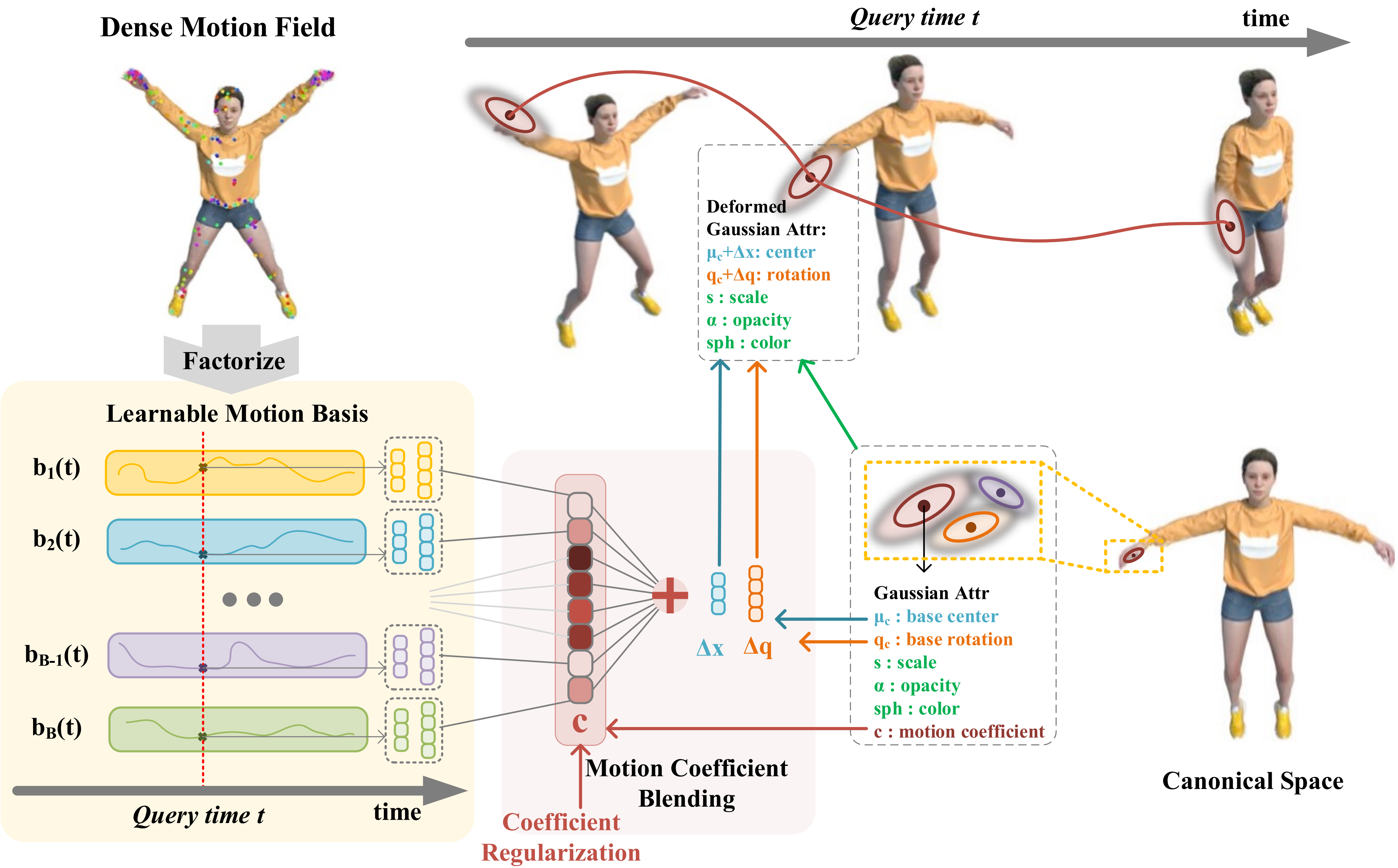
DynMF and MoSca



CaDeX and CaDeX++

DynMF: Neural Motion Factorization for Real-time Dynamic View Synthesis with 3D Gaussian Splatting

Agelos Kratimenos   Jiahui Lei   Kostas Daniilidis



Dense Motion Field
Query time t
time
Factorize
Learnable Motion Basis
b1(t)
b2(t)
bB-1(t)
bB(t)
Query time t
time
Deformed
Gaussian Attr:
μc+Δx: center
qc+Δq: rotation
s : scale
α : opacity
sph : color
Δx
Δq
c
Motion Coefficient
Blending
Coefficient
Regularization
Gaussian Attr
μc : base center
qc : base rotation
s : scale
α : opacity
sph : color
c : motion coefficient
Canonical Space

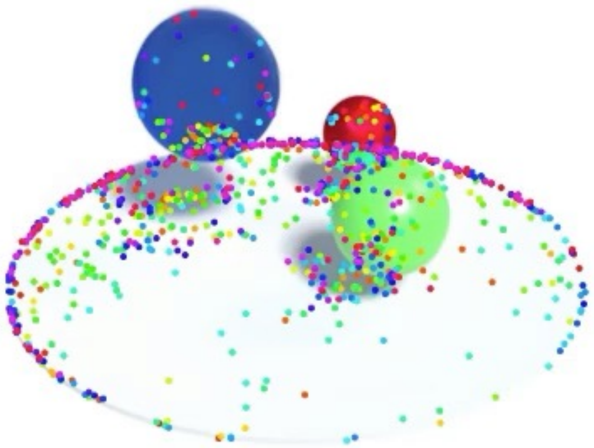
D-NeRF Dataset Results

Tracking

DynNeRF Dataset Results

DynNeRF Dataset Results

Ablation: L1 Loss



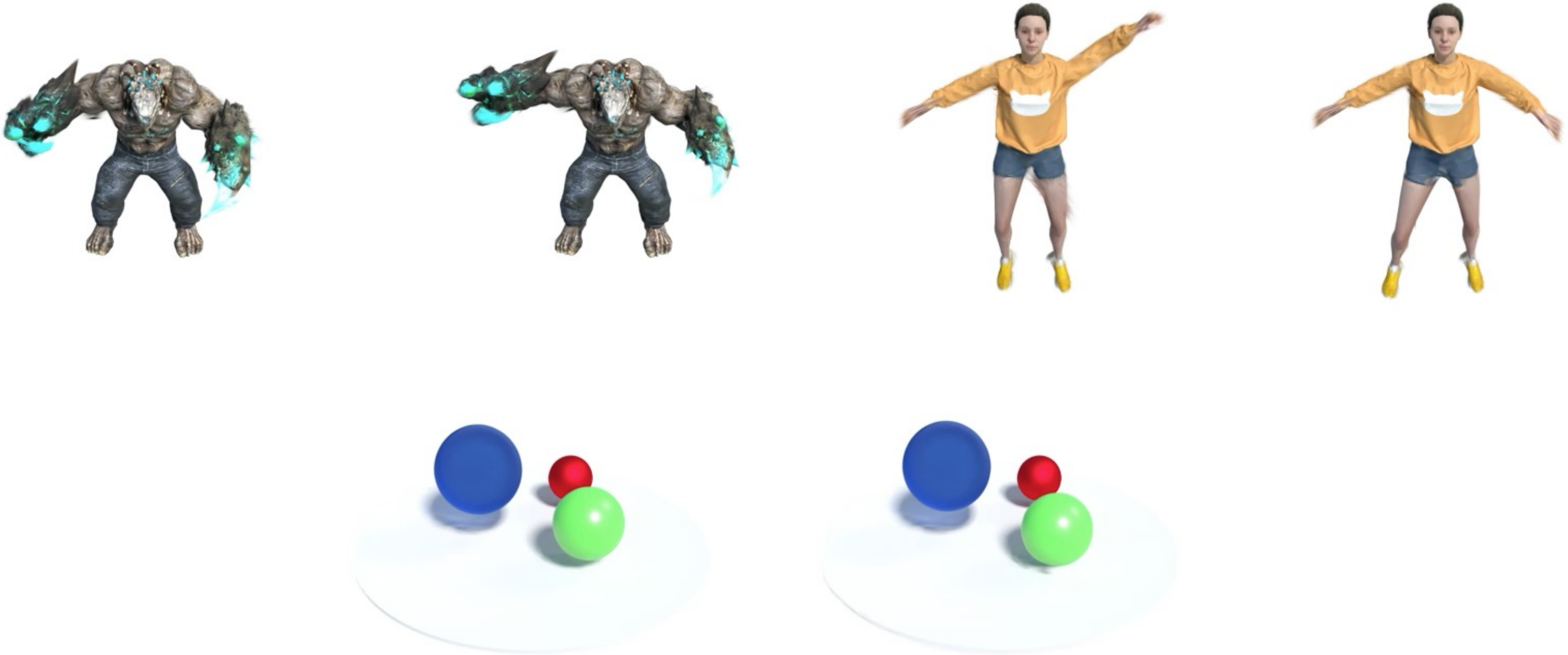
With L1Loss

Without L1Loss

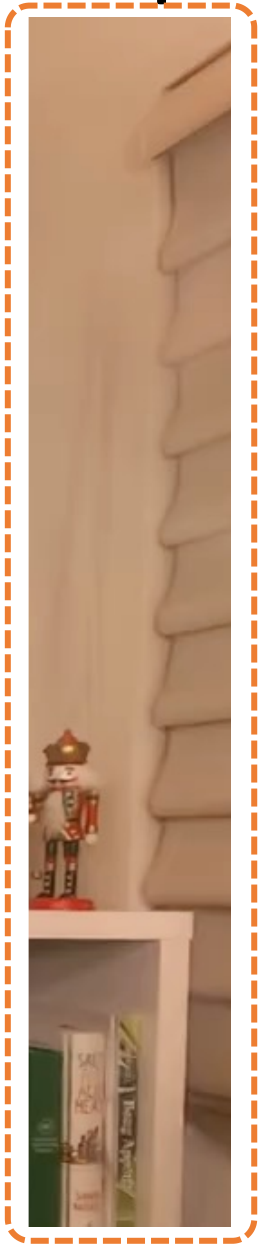
With L1Loss

Without L1Loss

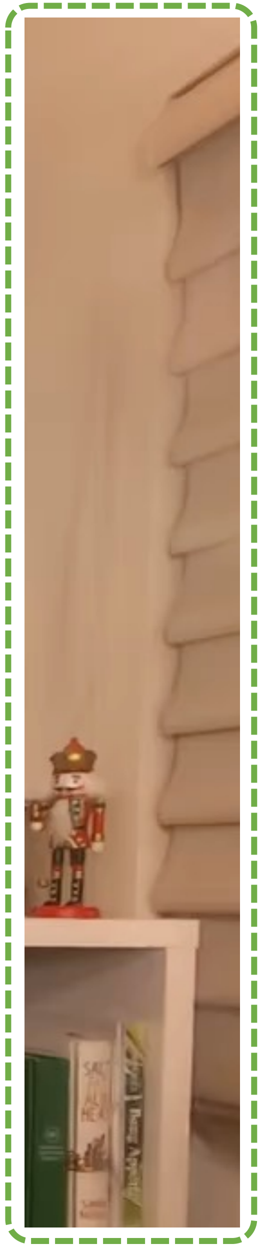
Motion Decomposition



Motion Decomposition Application



Wind in the background



Stabilized background motion

More to read: Shape of Motion

Shape of Motion: 4D Reconstruction from a Single Video

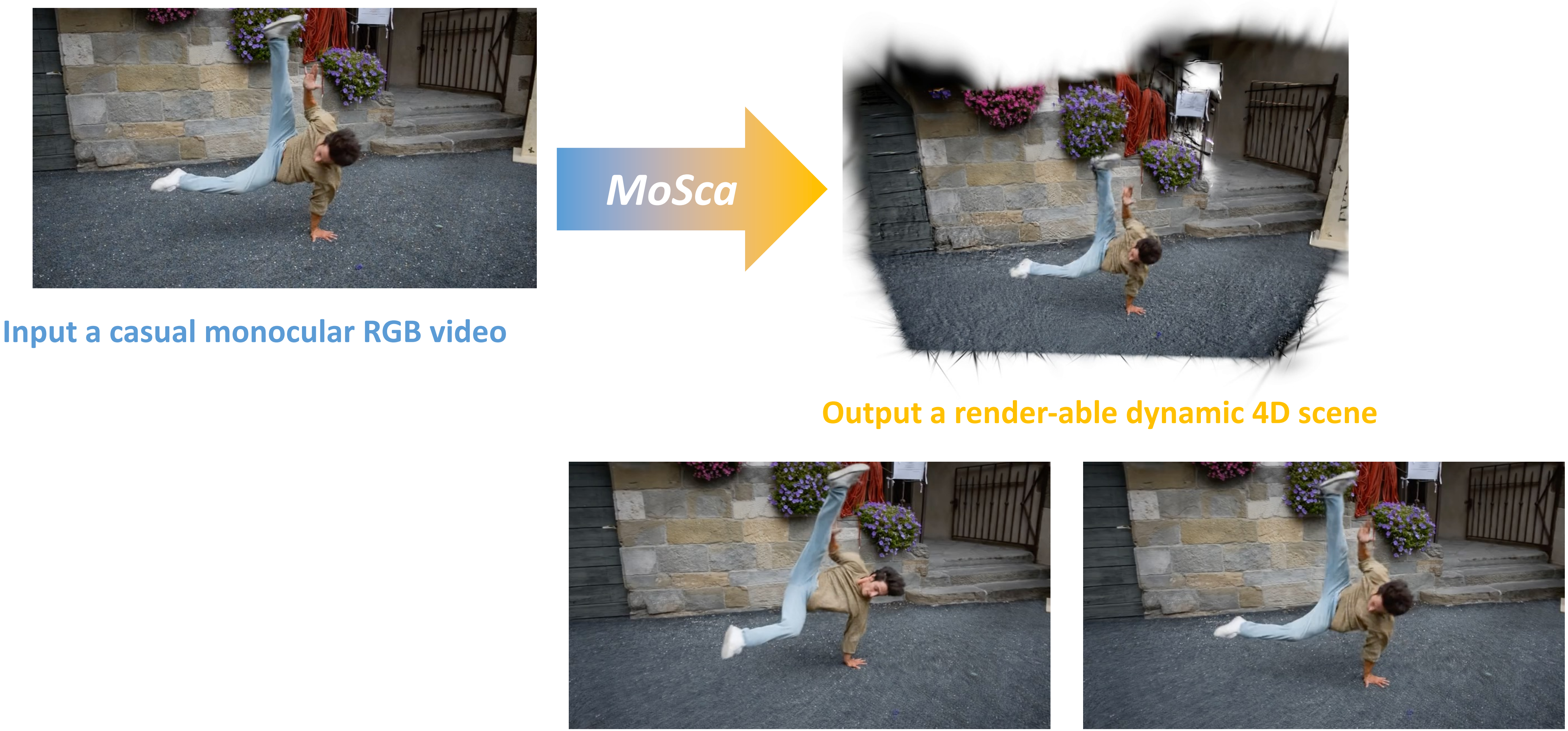
Qianqian Wang[1,2*], Vickie Ye[1*], Hang Gao[1*],
Jake Austin[1], Zhengqi Li[2], Angjoo Kanazawa[1]
[1]UC Berkeley [2]Google Research
* Equal Contribution

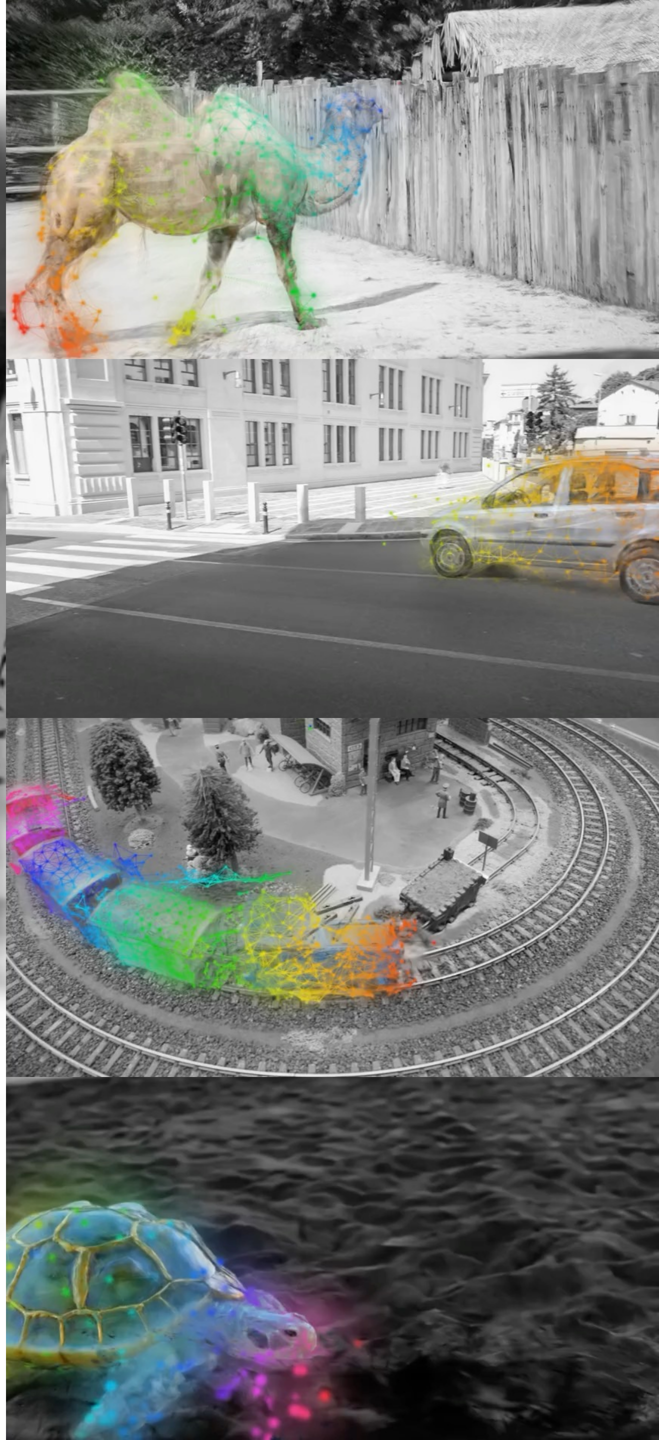
arXiv Code

Shape of Motion reconstructs a 4D scene from a single monocular video.

MoSca
Dynamic Gaussian Fusion from Casual Videos via 4D Motion Scaffolds

Jiahui Lei, Yijia Weng, Adam Harley,
Leonidas Guibas, Kostas Daniilidis

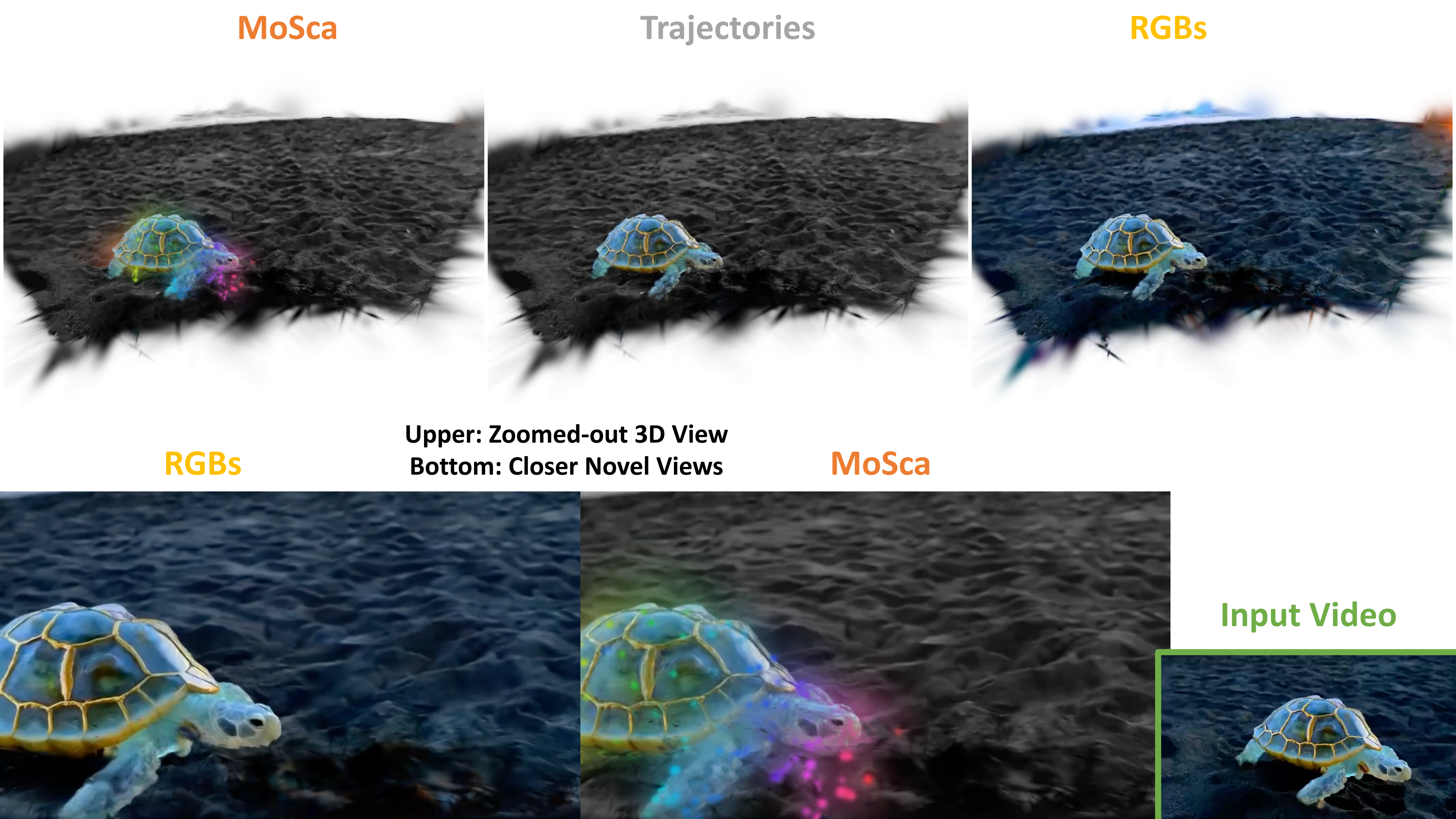

MoSca
Input a casual monocular RGB video
Output a render-able dynamic 4D scene



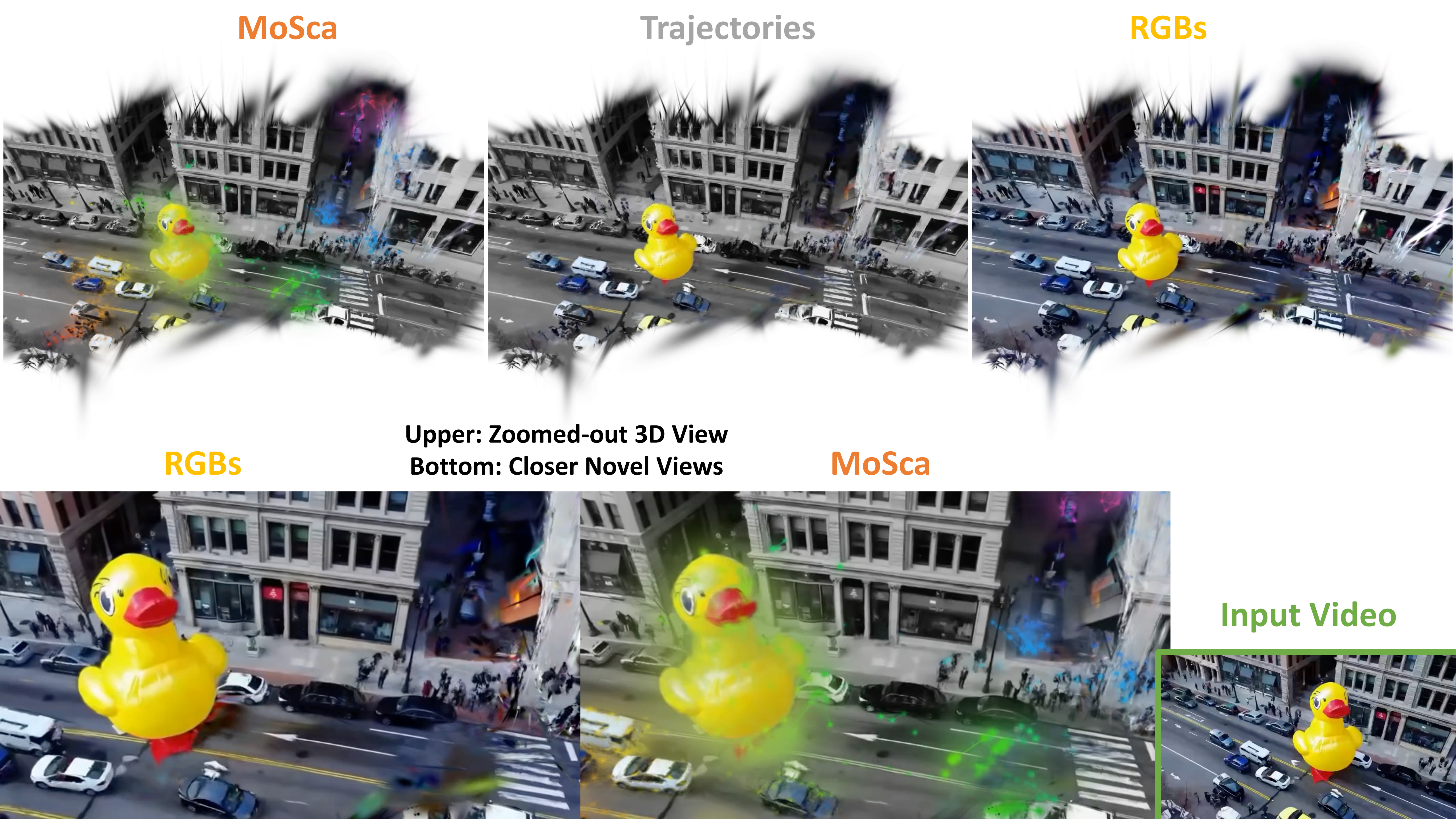
4D Motion Scaffolds

MoSca
Visual Results

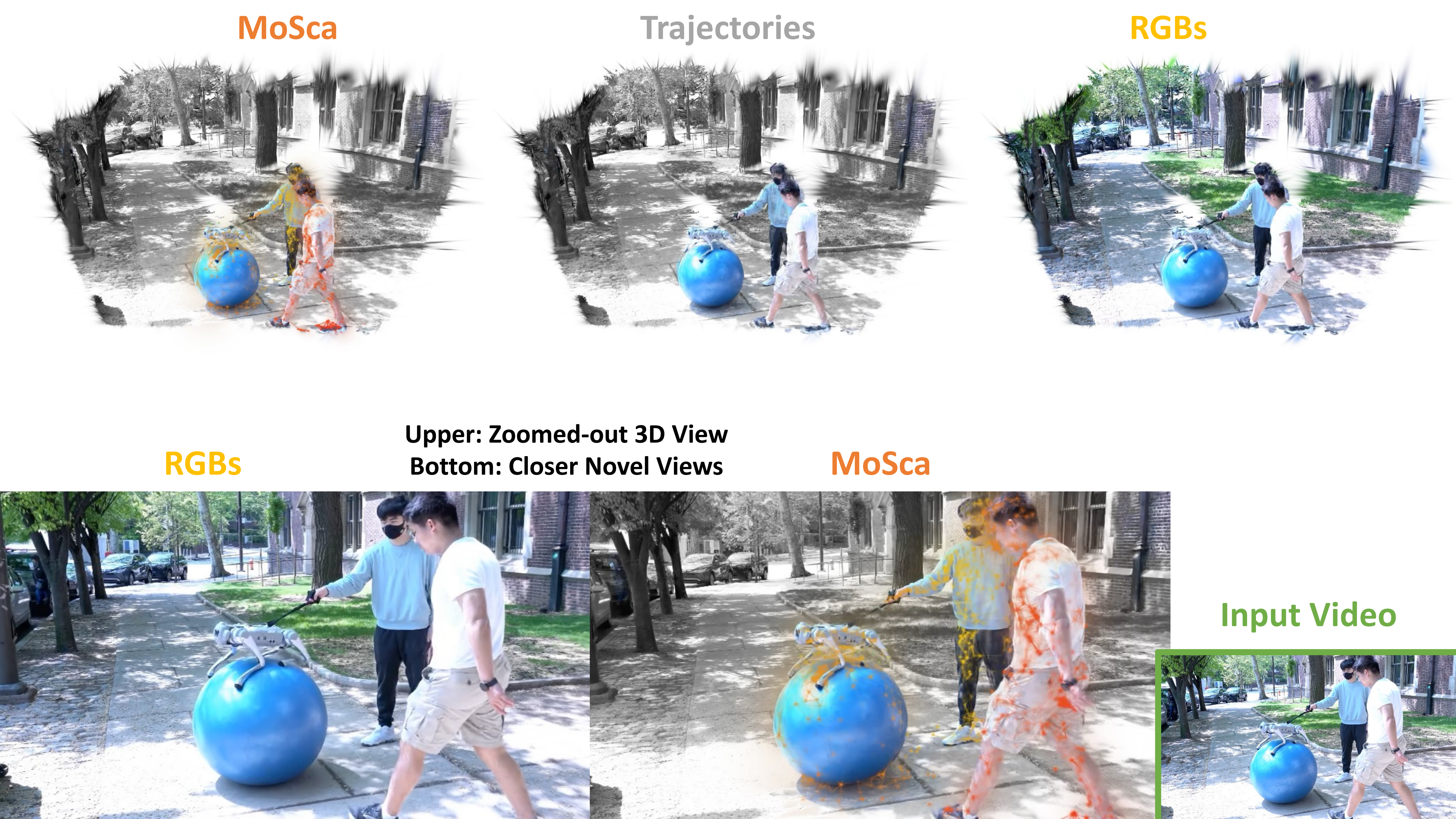
- OpenAI SORA Generated Videos
- Internet Videos of Robots
- Movie Clips
- DAVIS Dataset In-the-Wild Videos
- Comparison on iPhone DyCheck dataset
- Comparison on NVIDIA dataset


MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

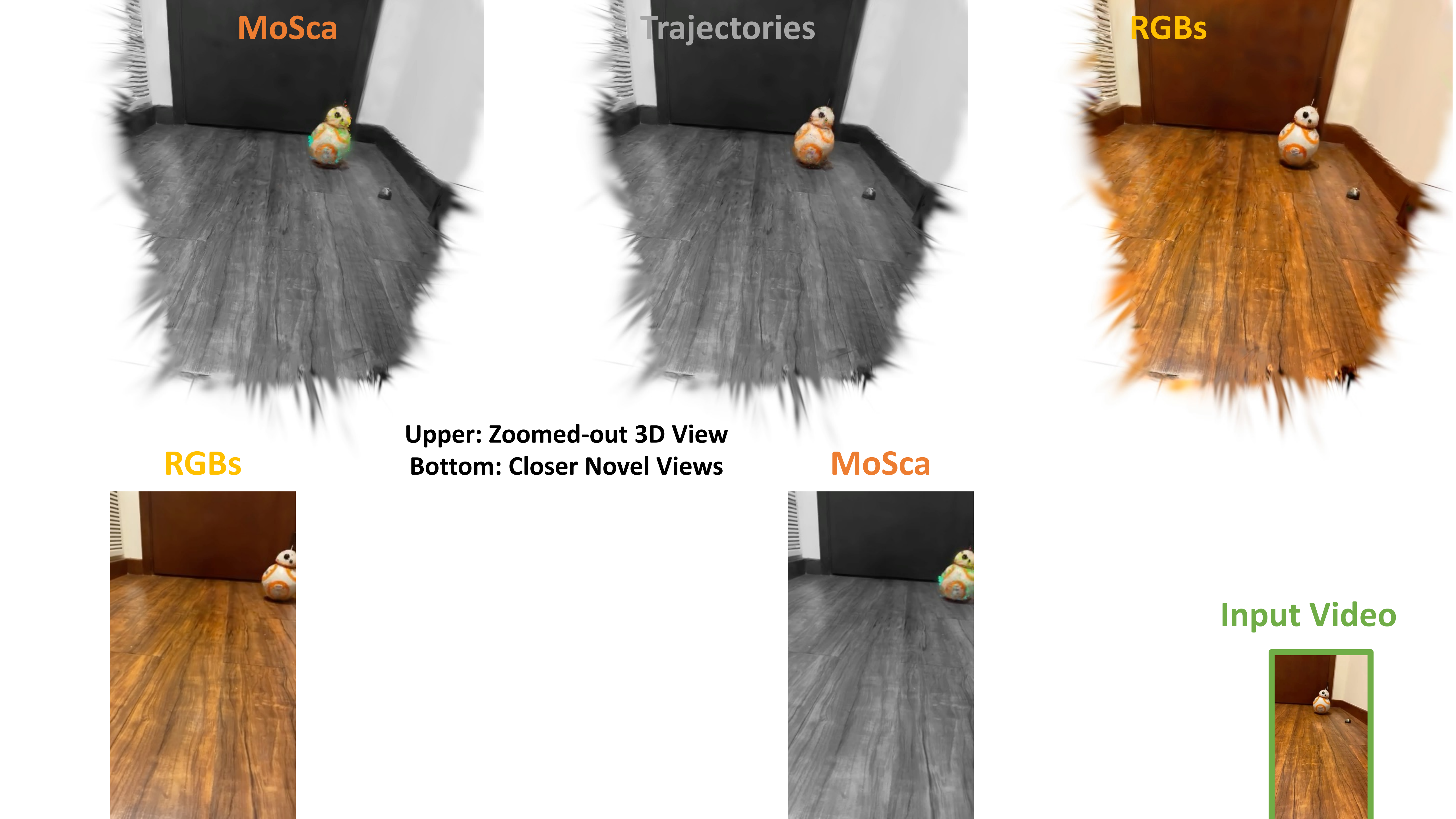

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

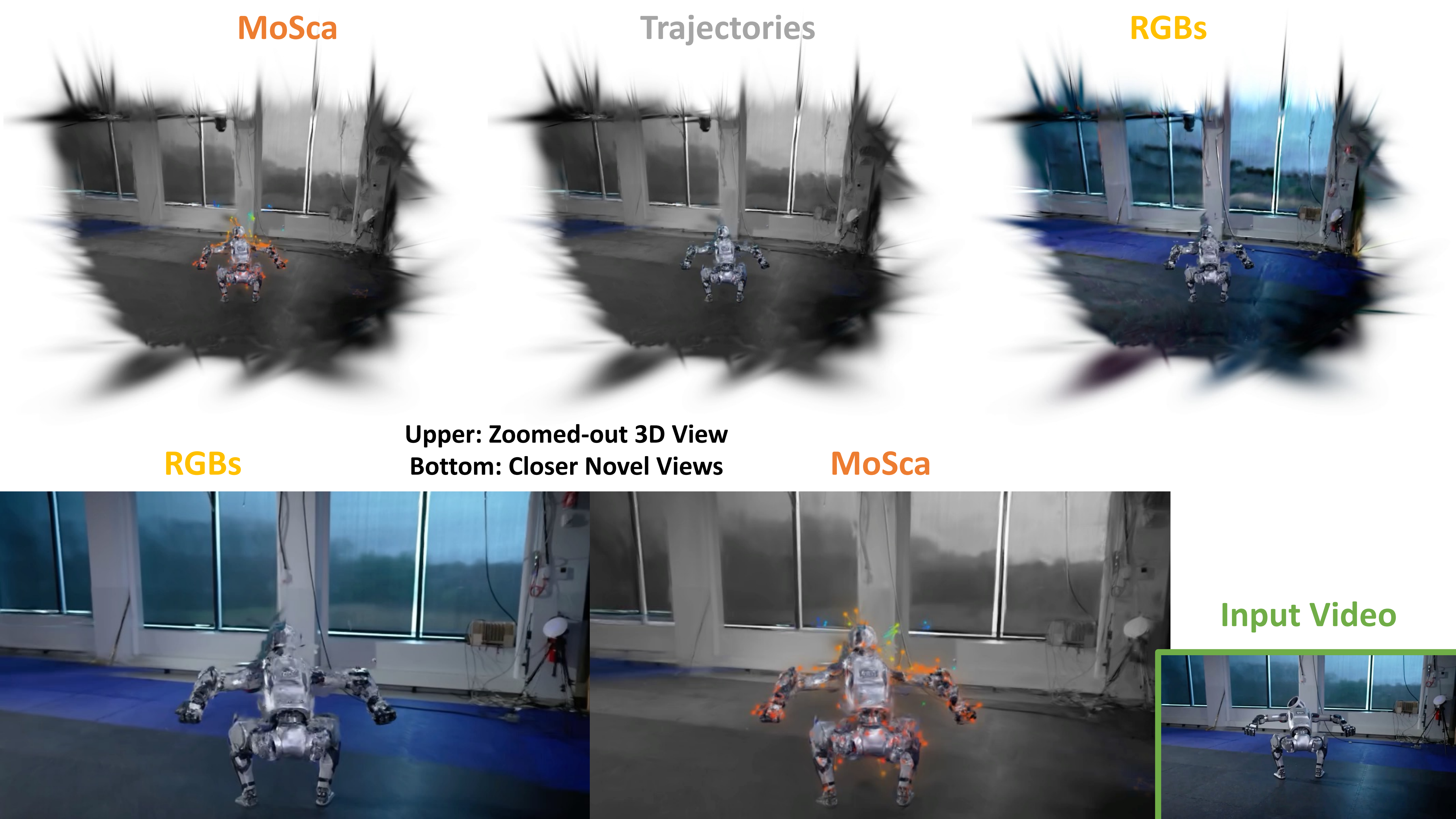

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video


MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

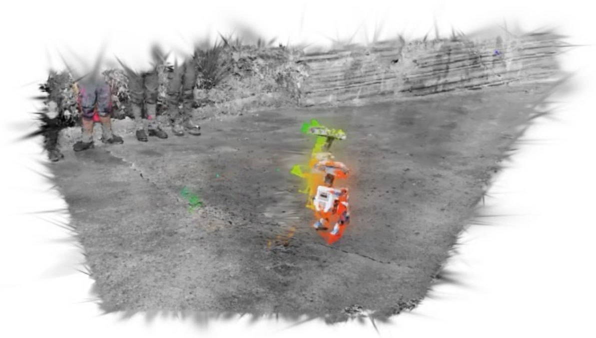


MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video


MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video


MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

MoSca



Trajectories

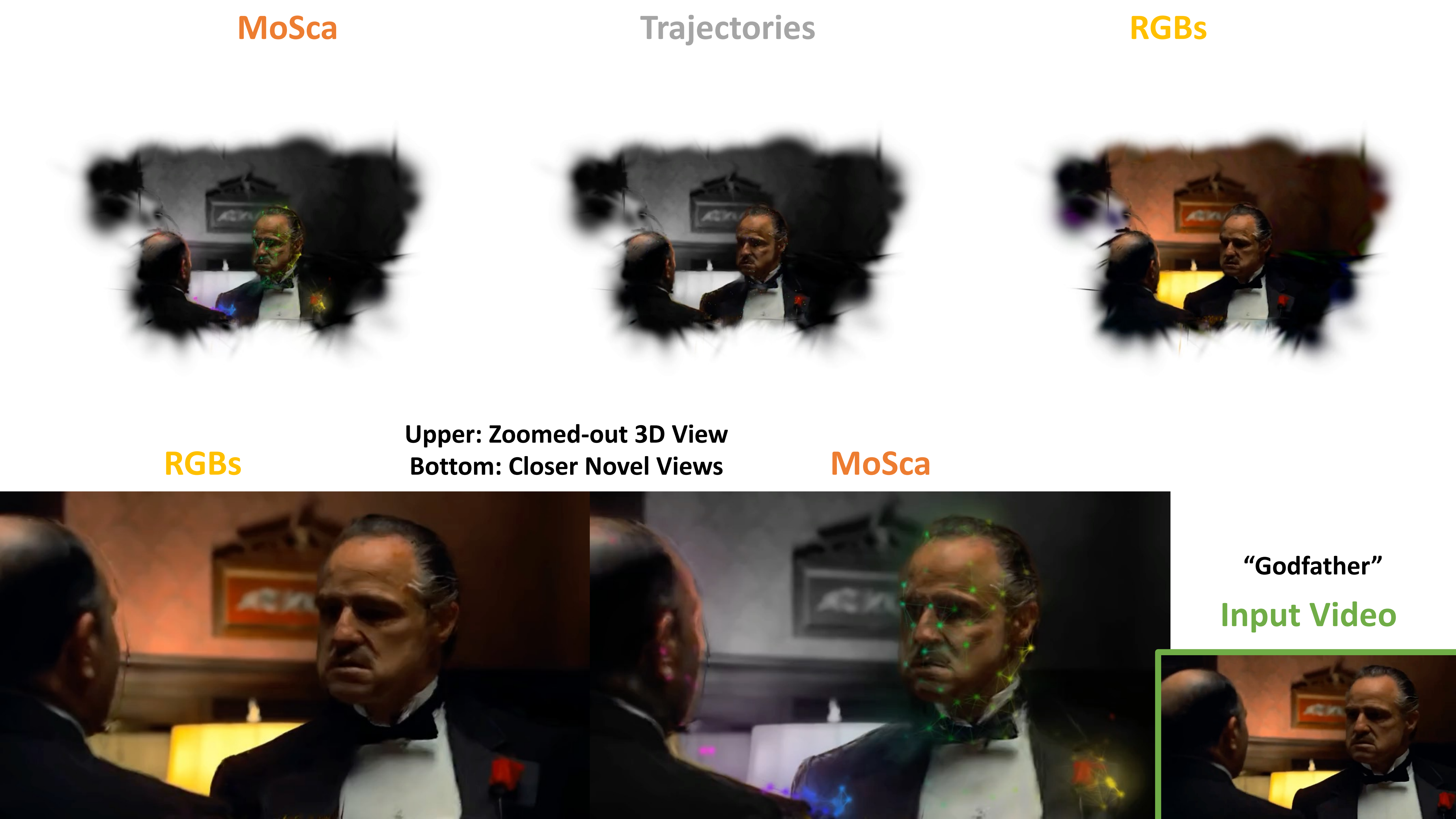
RGBs

Upper: Zoomed-out 3D View
Bottom: Closer Novel Views

RGBs

MoSca

Input Video


MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
“Godfather”
Input Video


MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
"Mr. Bean"
Input Video

MoSca

Trajectories

RGBs

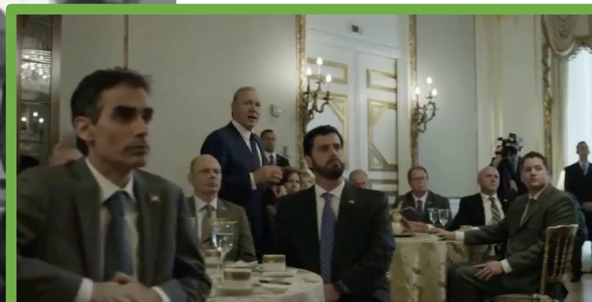
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views

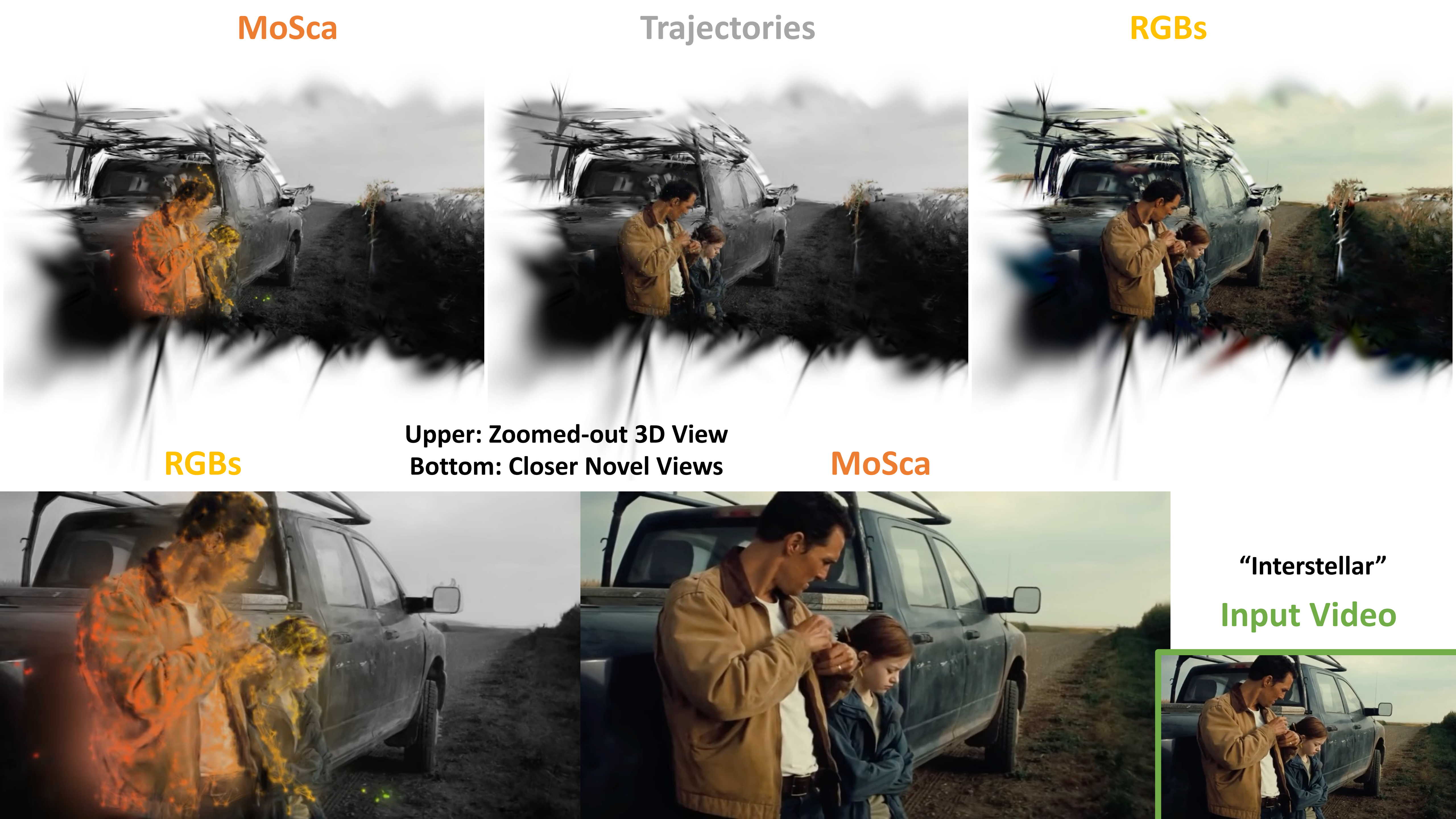
RGBs

MoSca

“House of Cards”

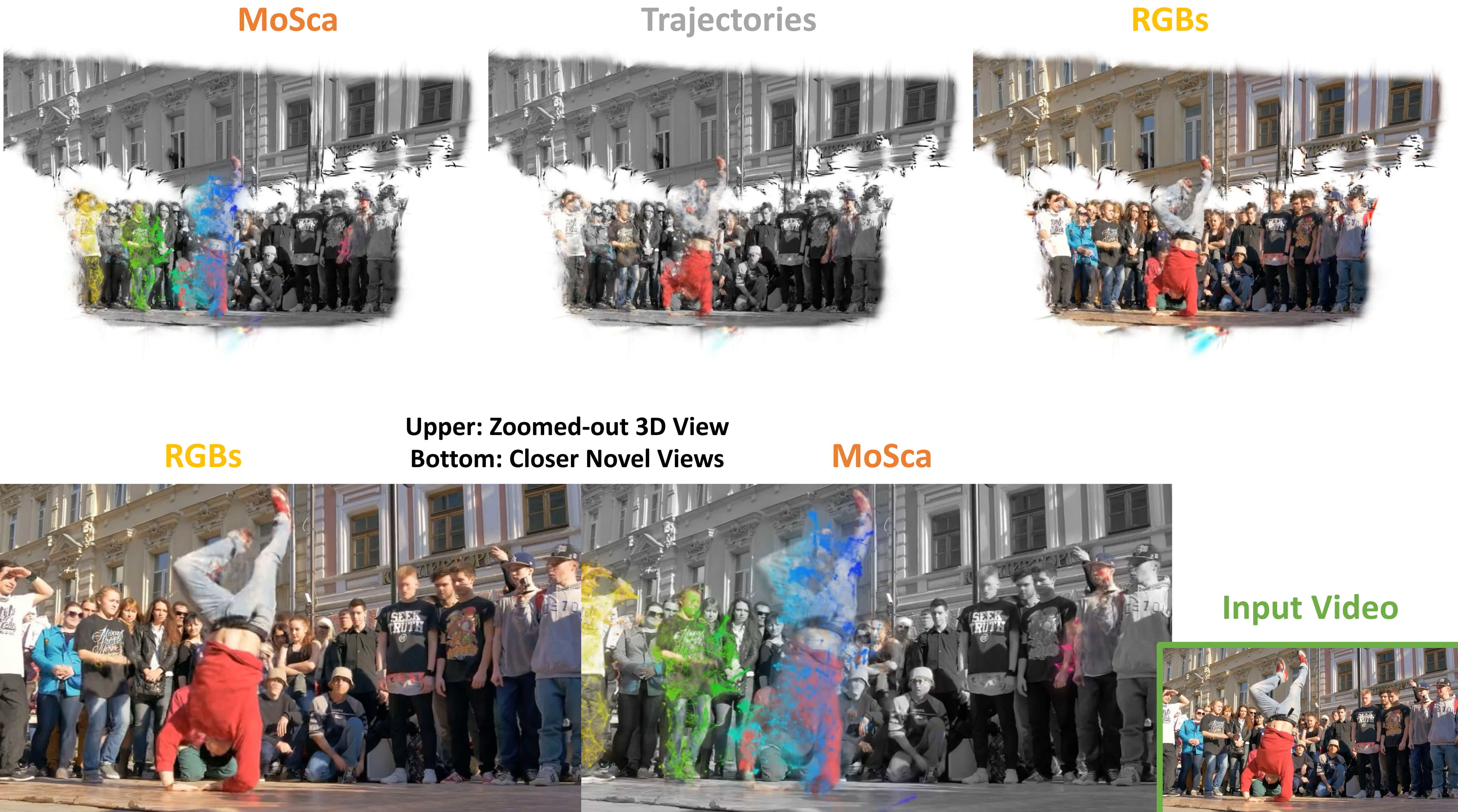
Input Video



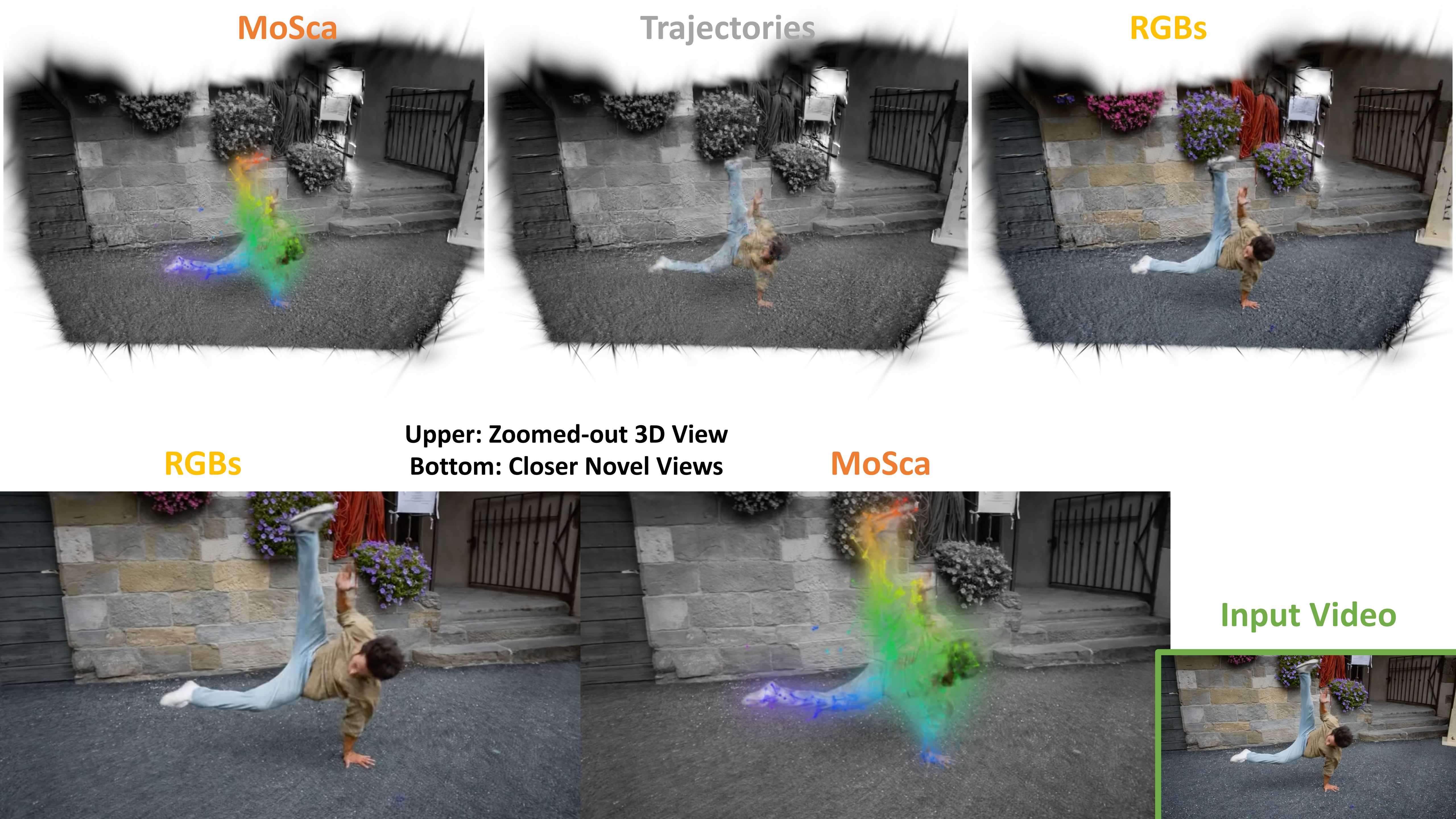

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
“Interstellar”
Input Video

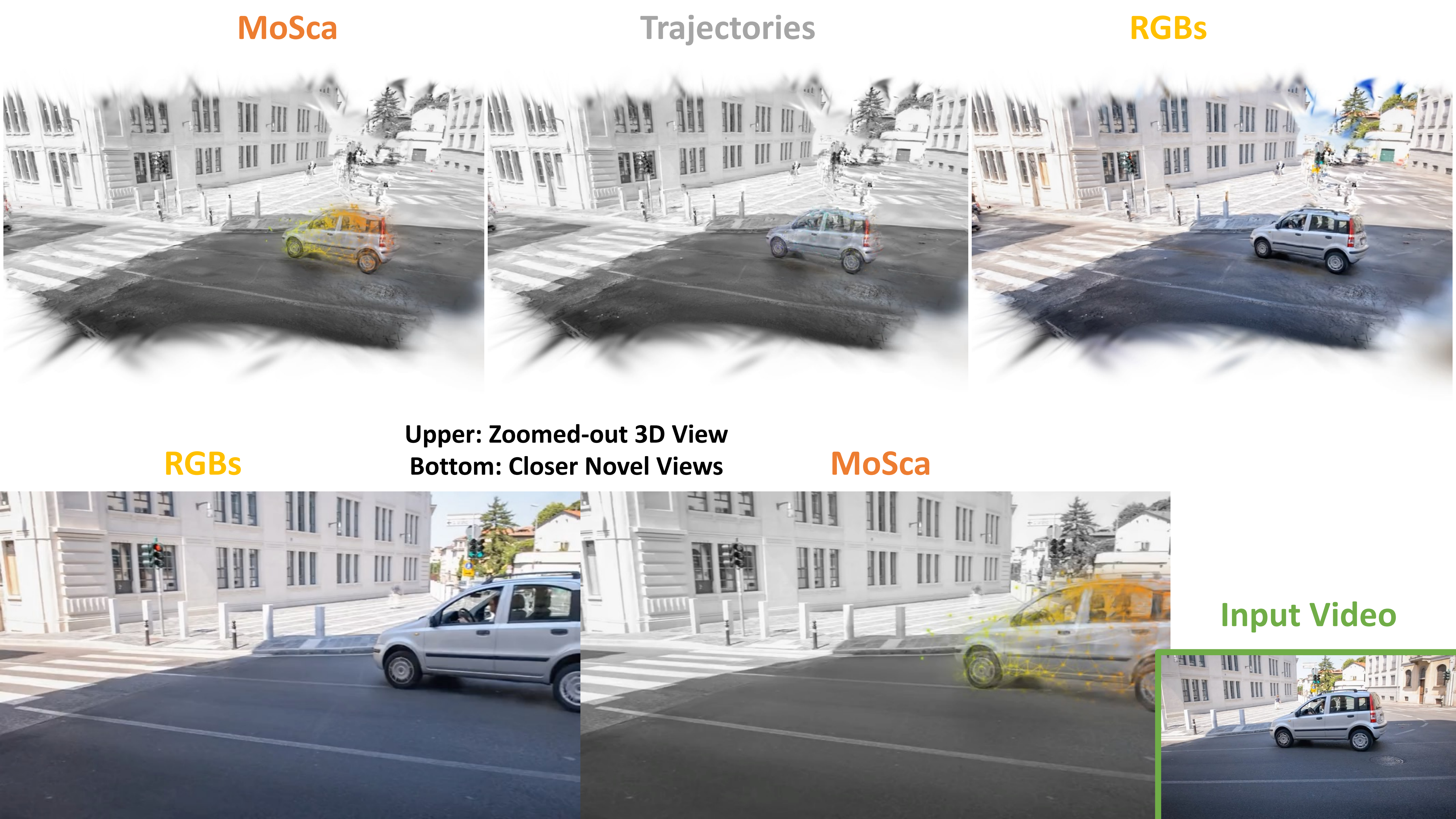

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

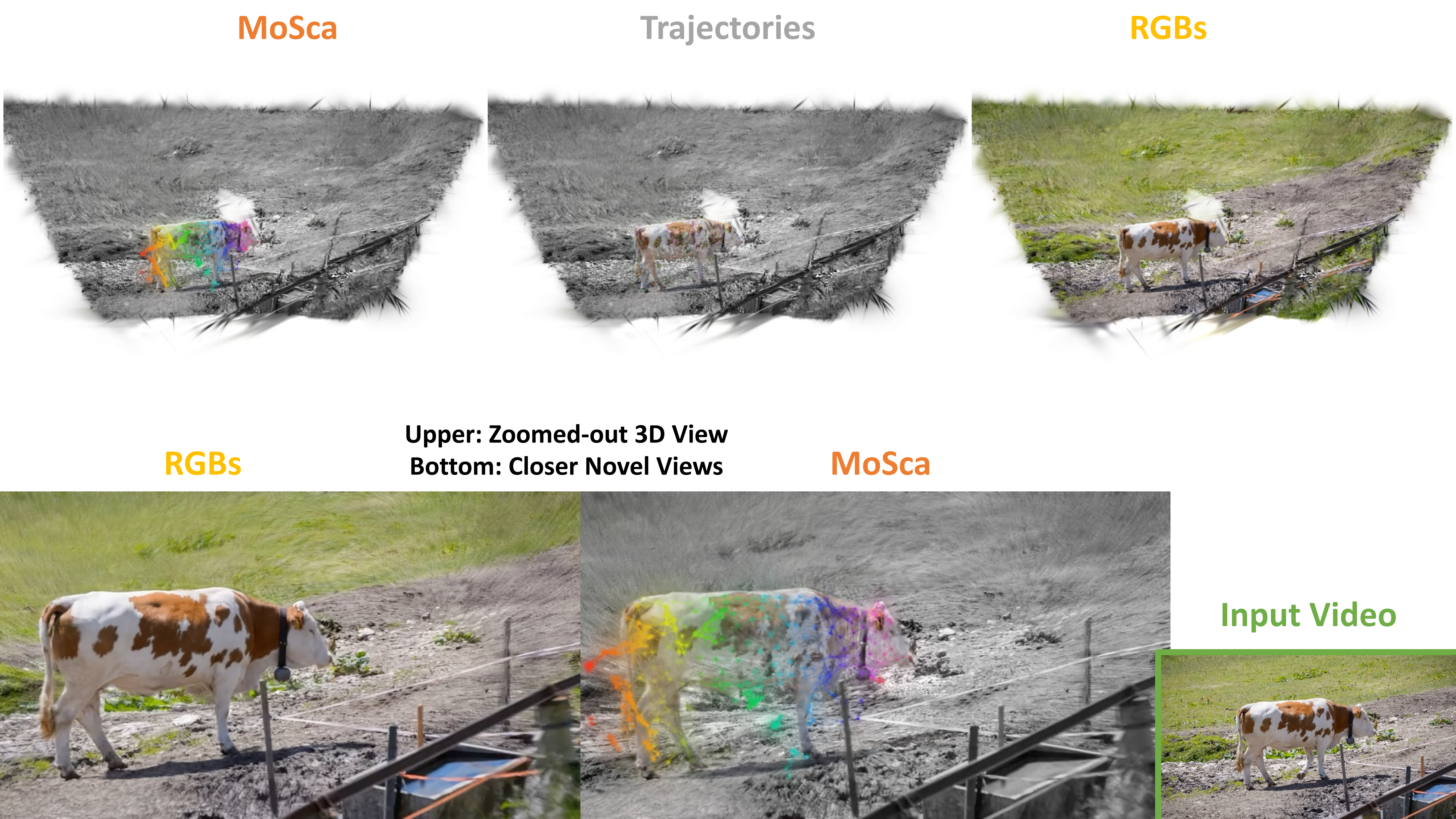

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

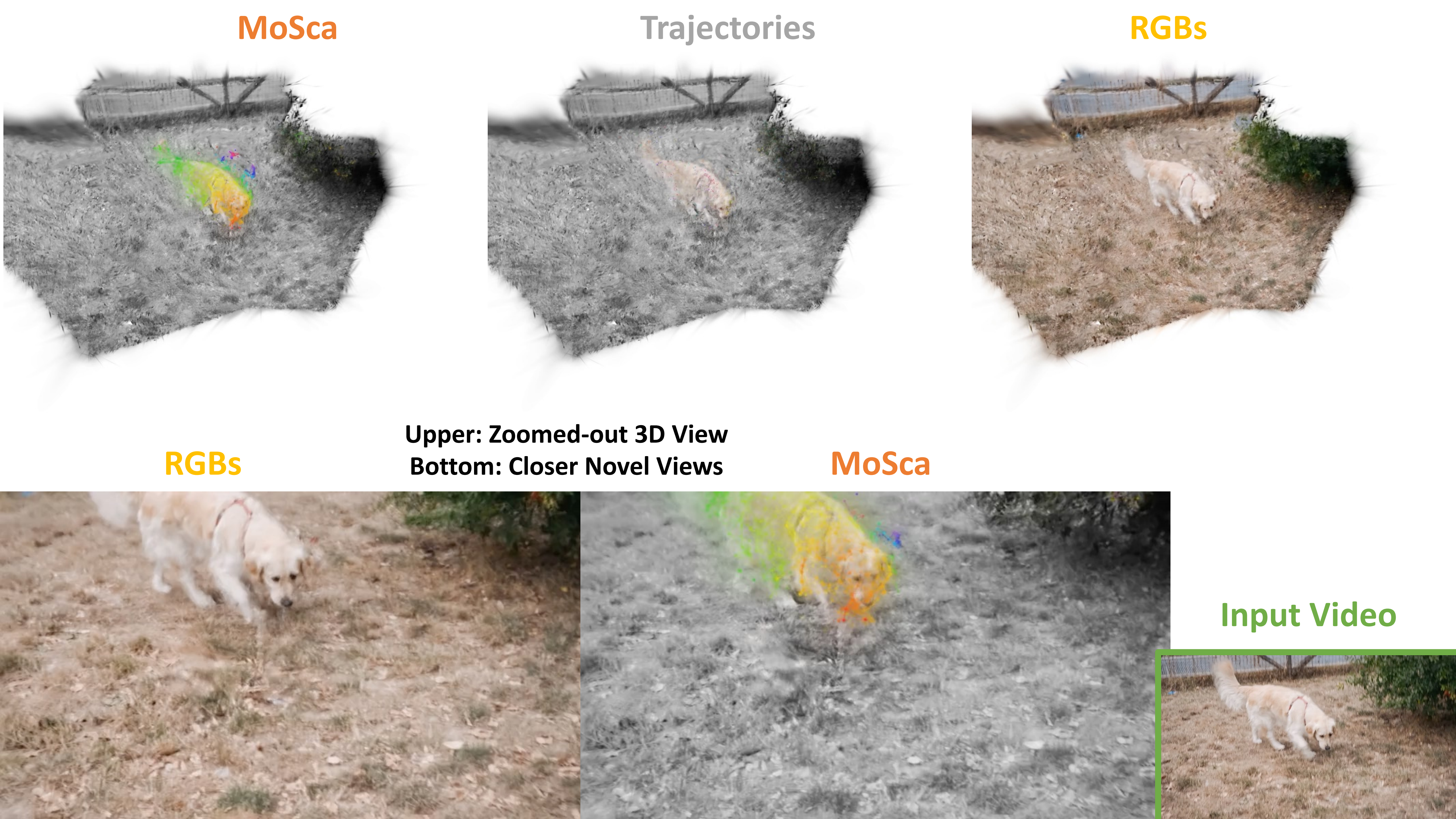


MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

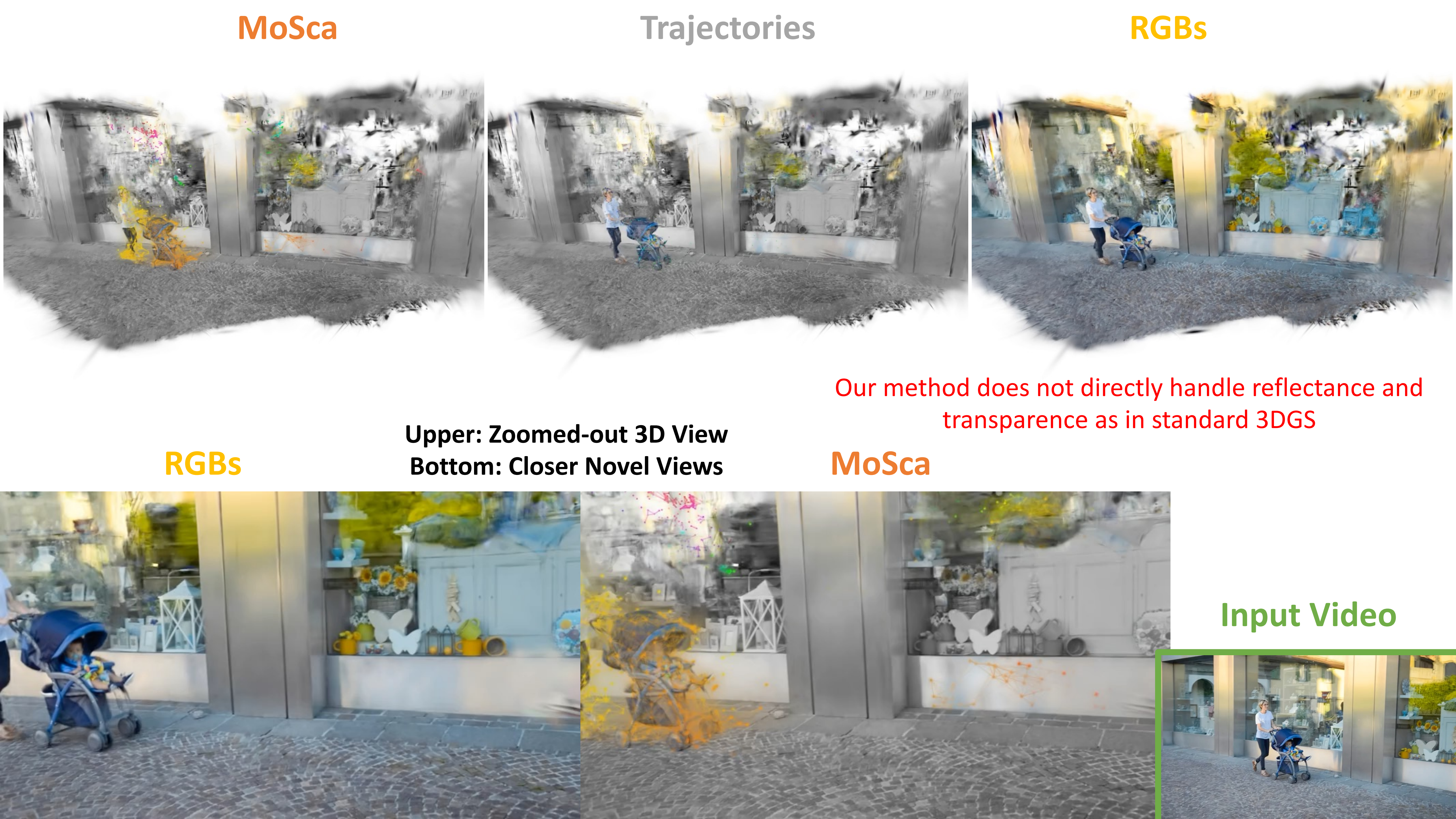

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

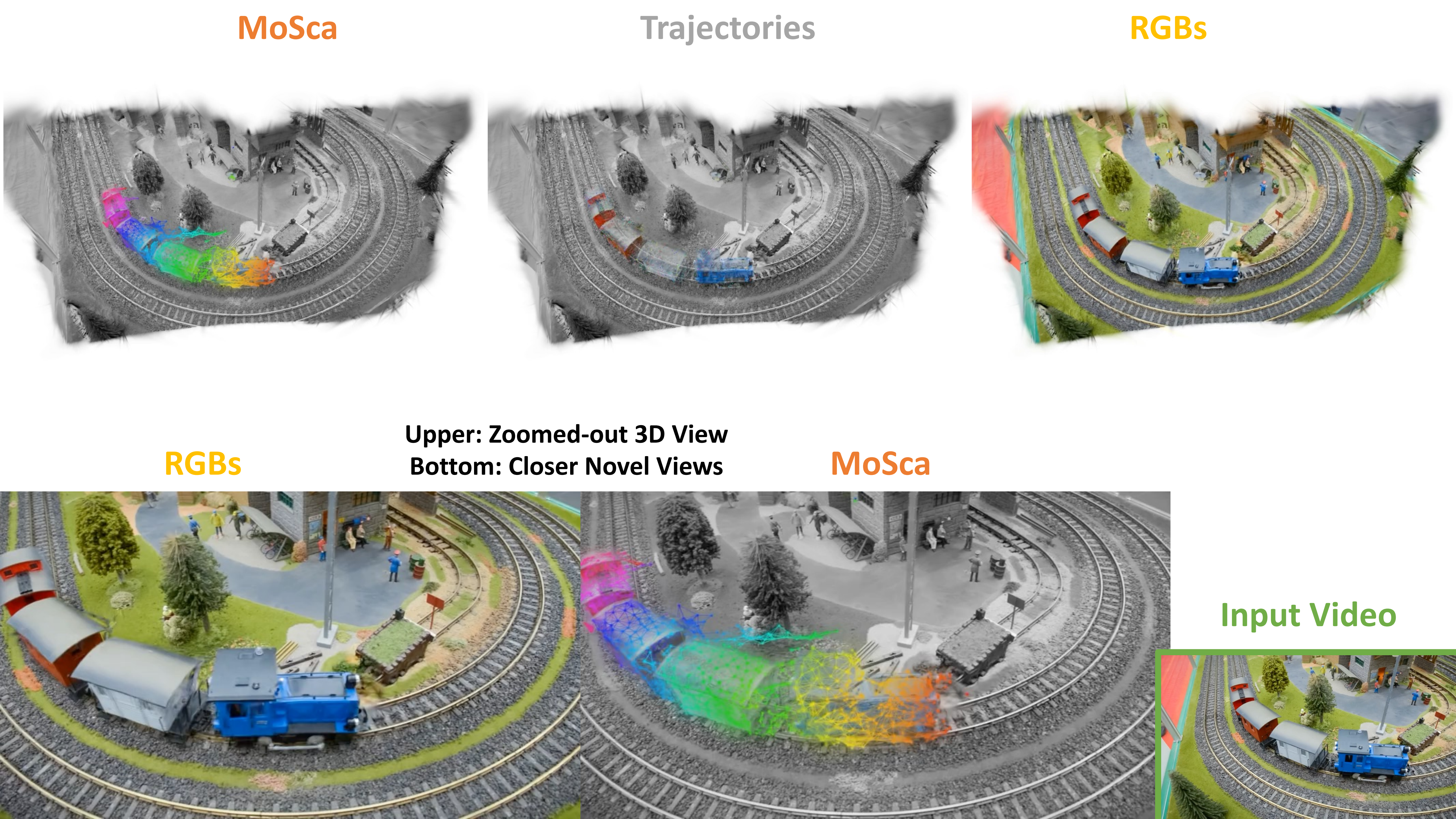

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

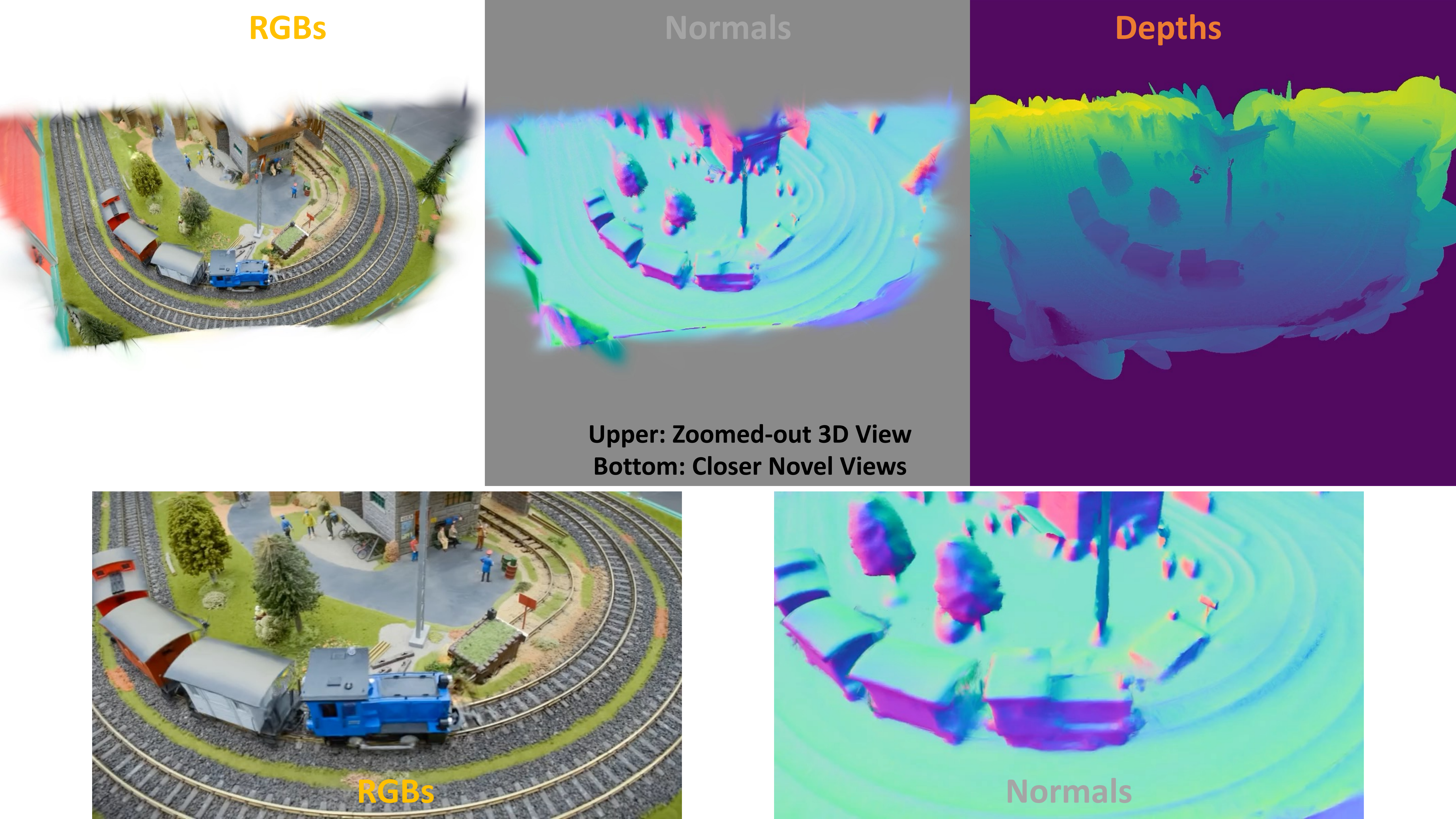

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

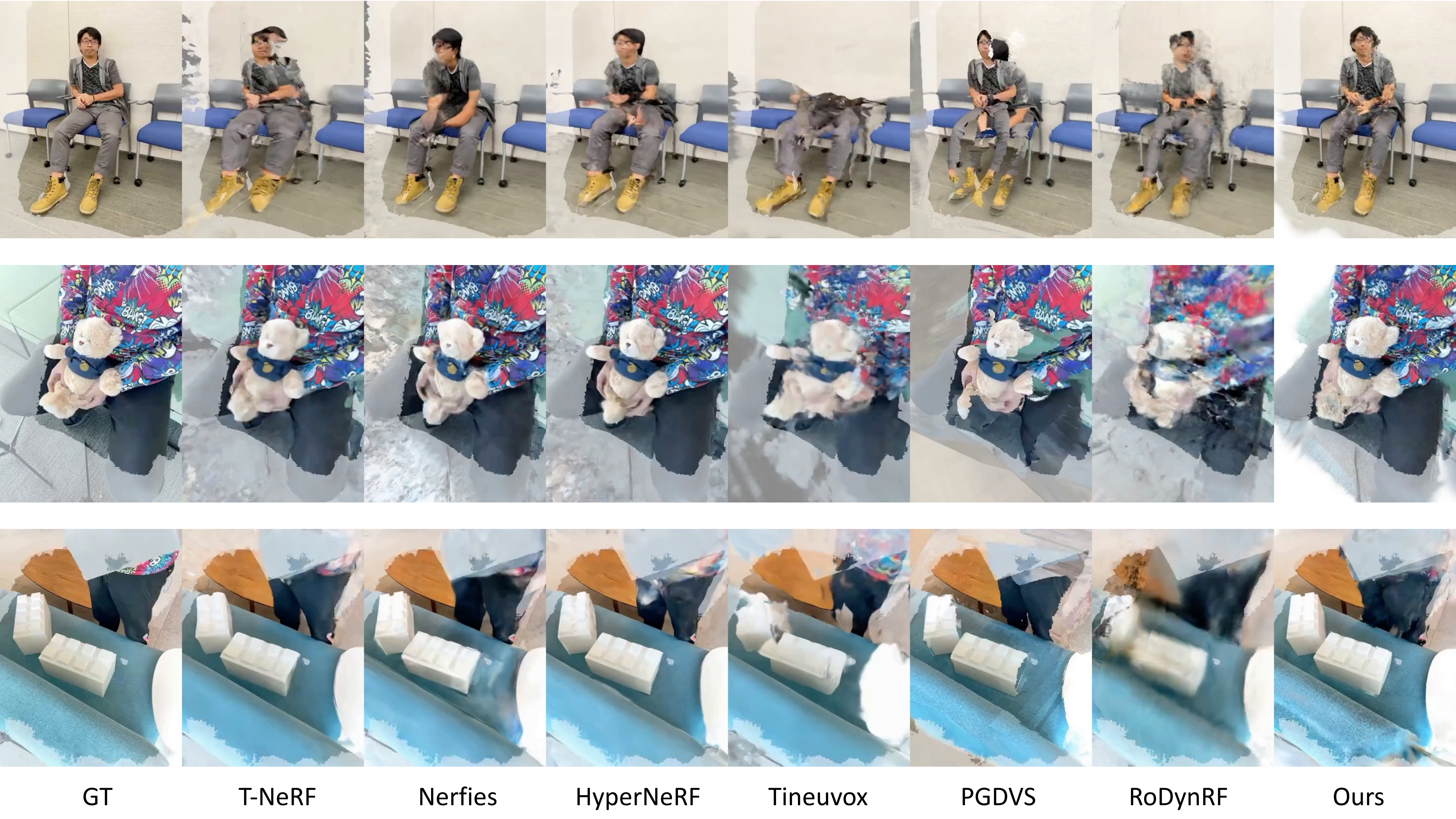

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video


MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

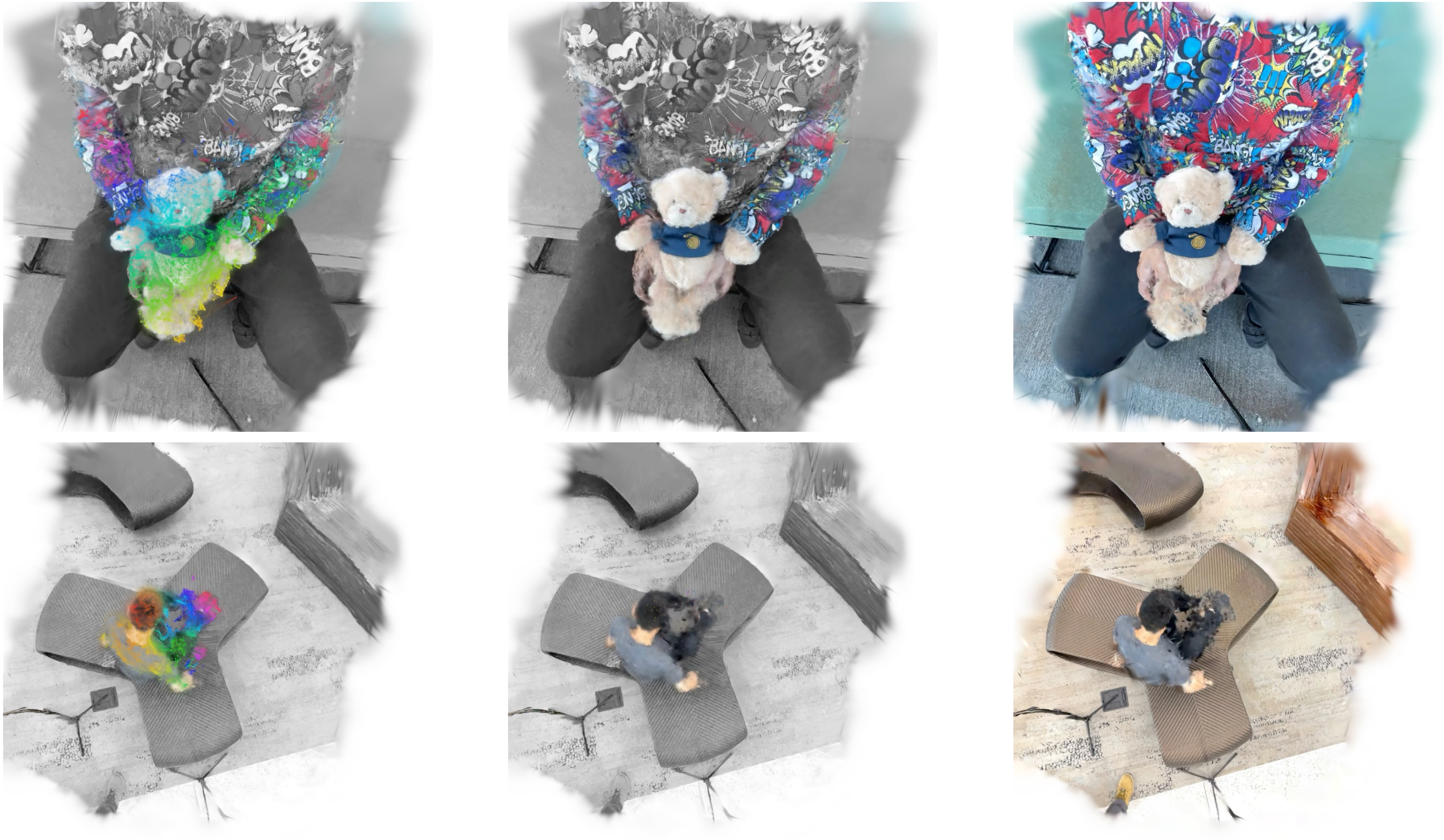

MoSca
Trajectories
RGBs
Our method does not directly handle reflectance and transparence as in standard 3DGS
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video

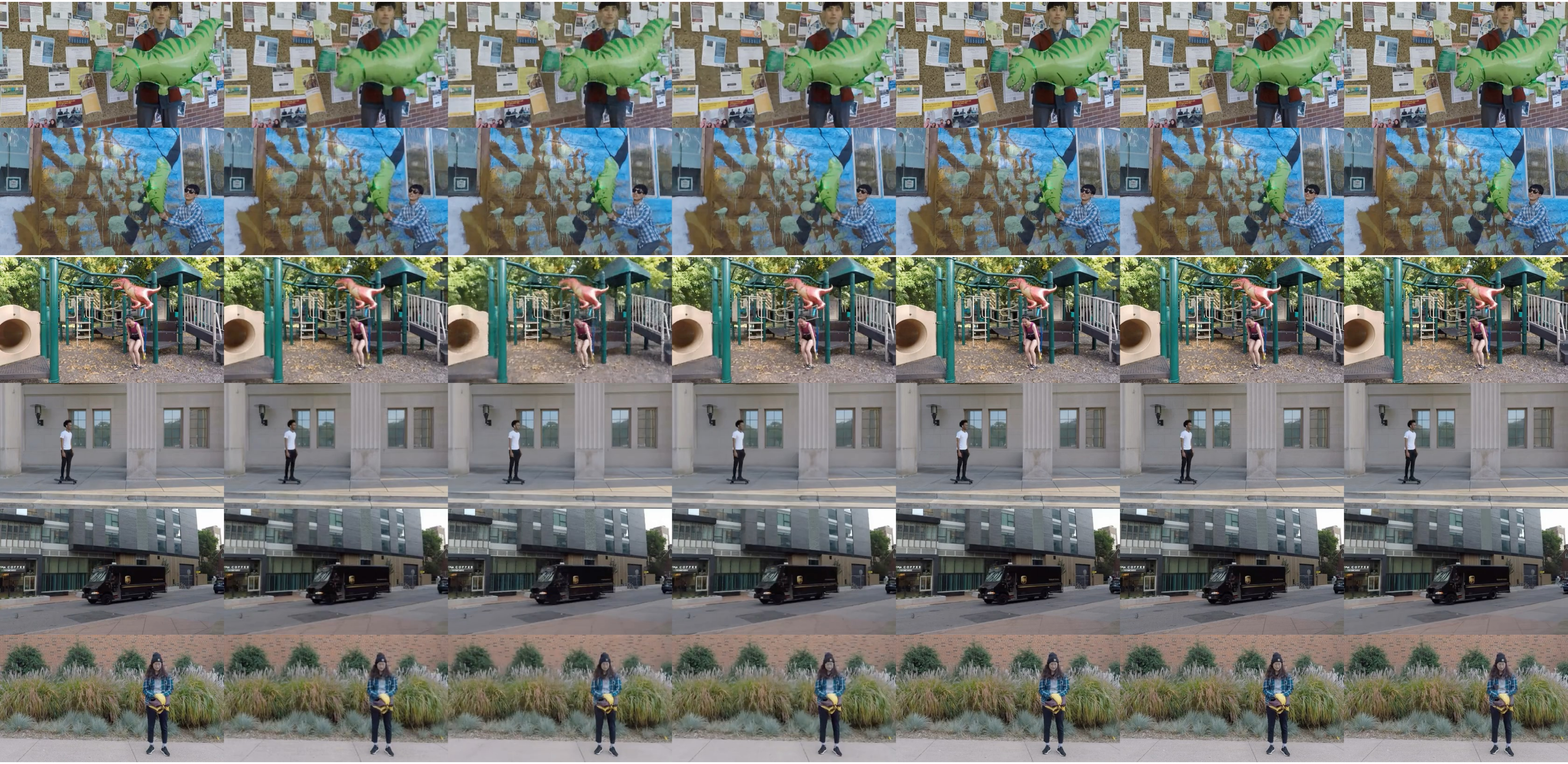

MoSca
Trajectories
RGBs
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
MoSca
Input Video


RGBs
Normals
Depths
Upper: Zoomed-out 3D View
Bottom: Closer Novel Views
RGBs
Normals


GT
T-NeRF
Nerfies
HyperNeRF
Tineuvox
PGDVS
RoDynRF
Ours

GT T-NeRF Nerfies HyperNeRF Tineuvox PGDVS RoDynRF Ours



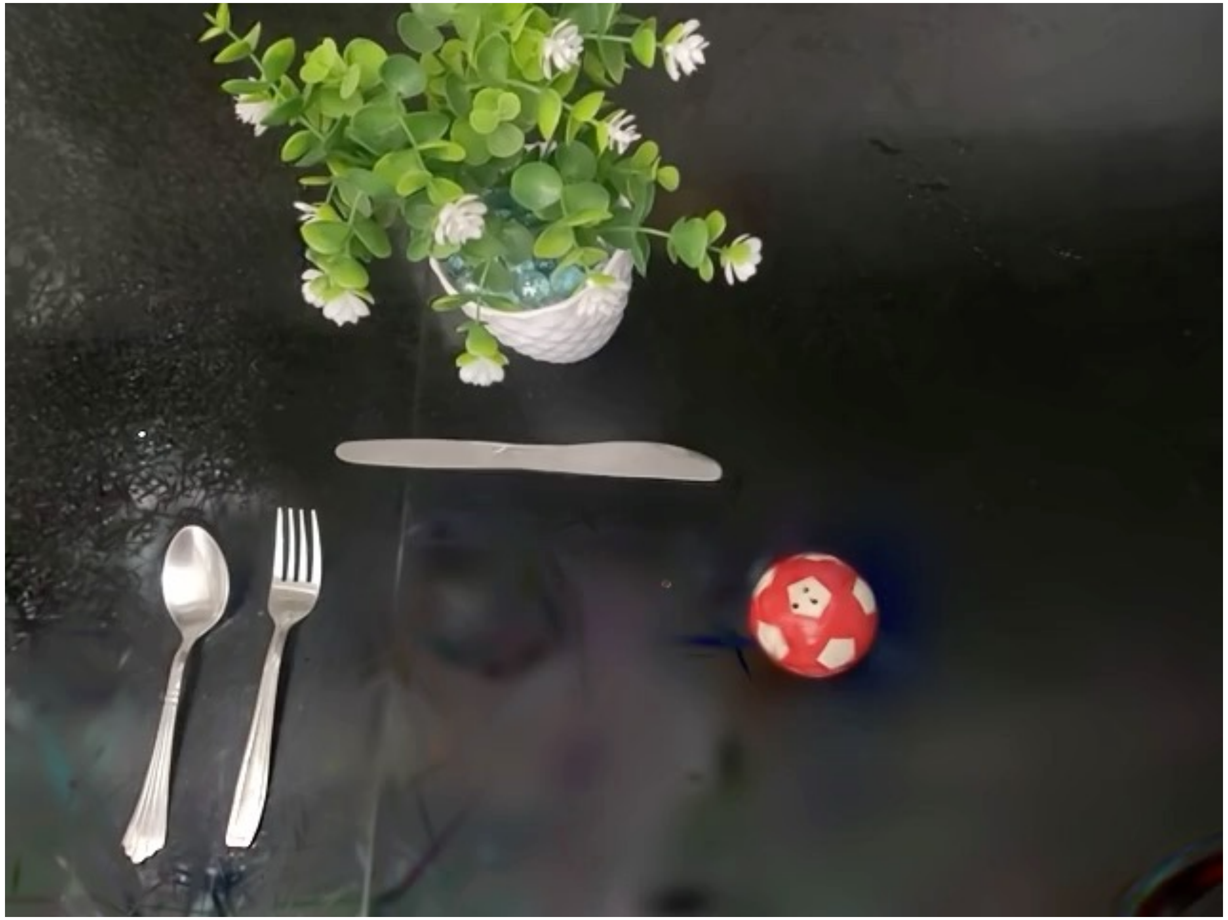


GT NSFF HyperNeRF Tineuvox DynamicNeRF RoDynRF COLMAP Free Ours COLMAP Free

MoSca: ARAP & Rendering may hide the ball

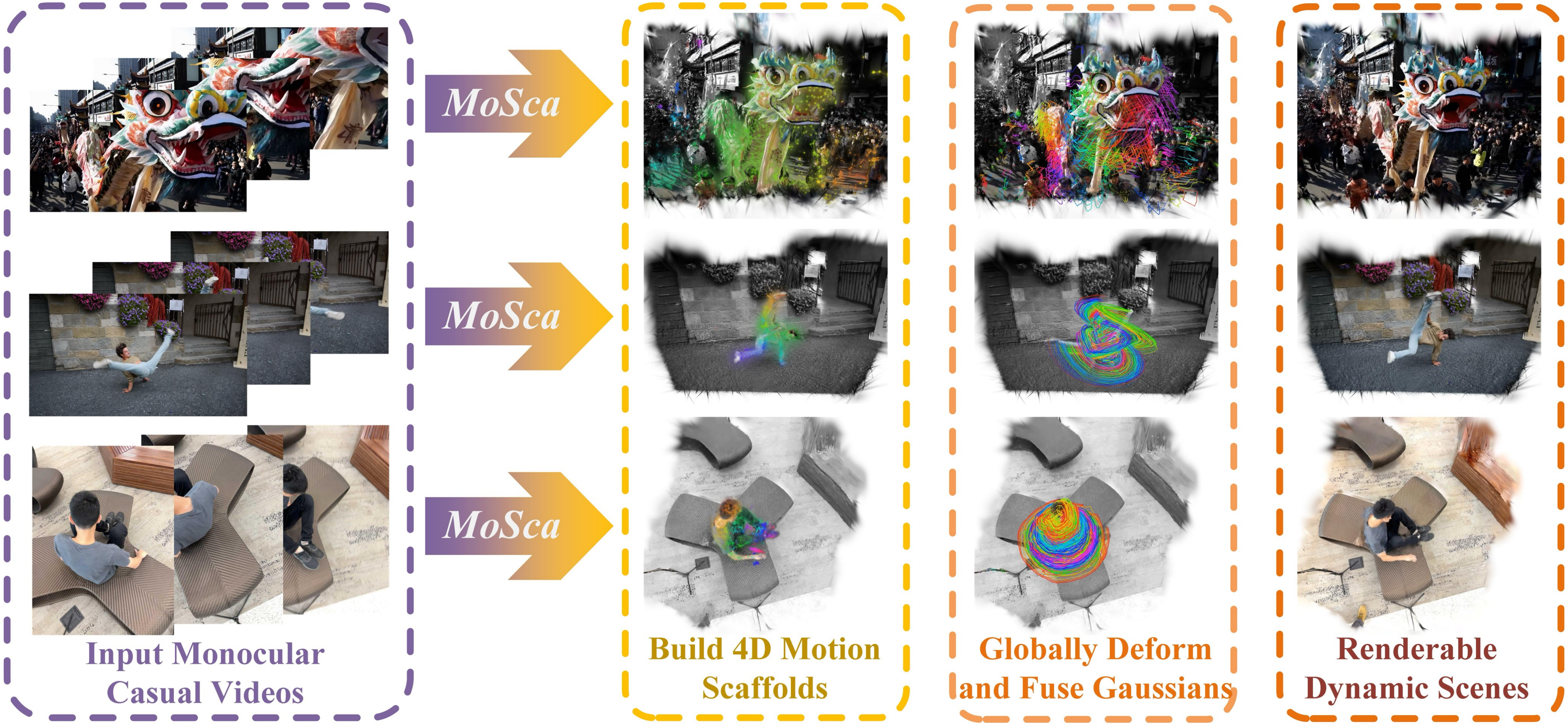
Input video

Recon input view

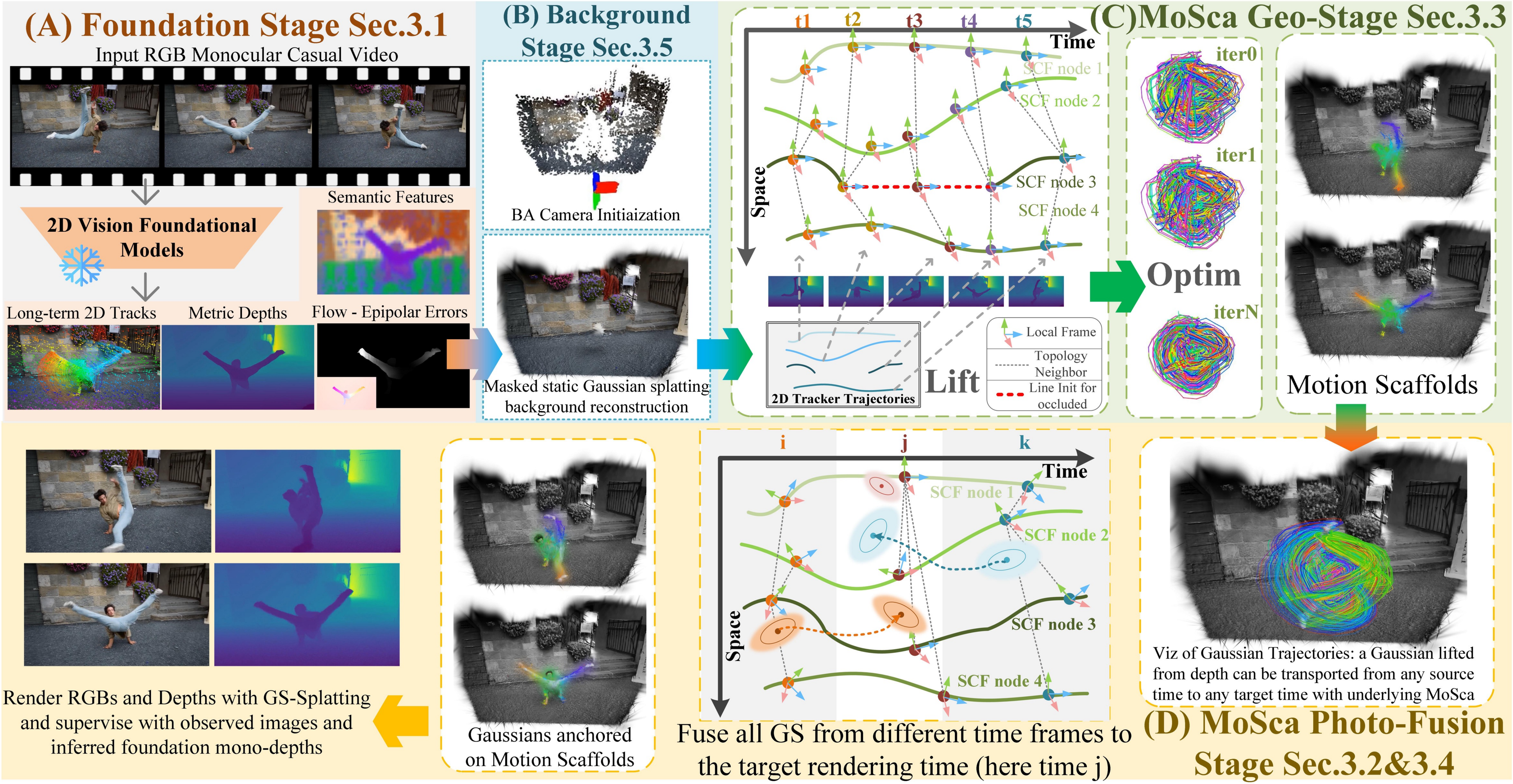


Look through, only show ball

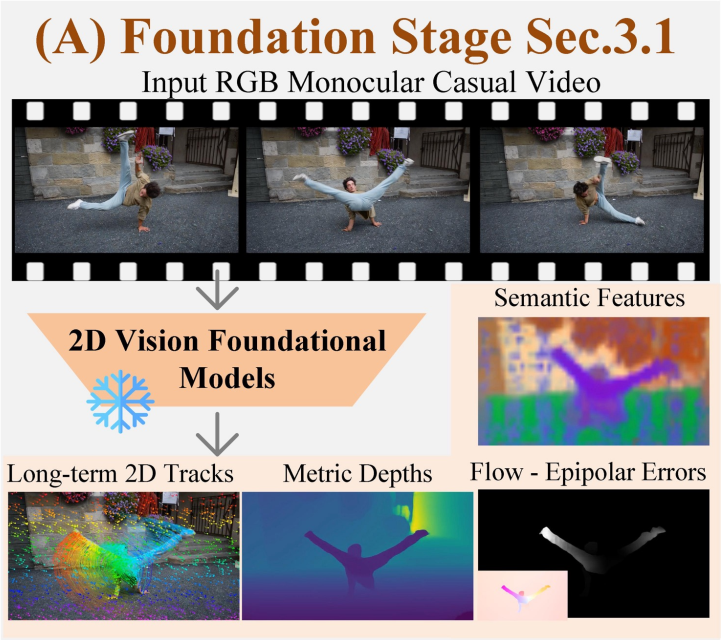
How it works?



MoSca
MoSca
MoSca
Input Monocular
Casual Videos
Build 4D Motion
Scaffolds
Globally Deform
and Fuse Gaussians
Renderable
Dynamic Scenes



(A) Foundation Stage Sec.3.1
Input RGB Monocular Casual Video
2D Vision Foundational Models
Semantic Features
Long-term 2D Tracks
Metric Depths
Flow - Epipolar Errors
(B) Background Stage Sec.3.5
BA Camera Initiaization
Masked static Gaussian splatting background reconstruction
(C)MoSca Geo-Stage Sec.3.3
t1
t2
t3
t4
t5
Time
Space
SCF node 1
SCF node 2
SCF node 3
SCF node 4
2D Tracker Trajectories
Lift
Local Frame
Topology Neighbor
Line Init for occluded
iter0
iter1
iterN
Optim
Motion Scaffolds
Render RGBs and Depths with GS-Splatting and supervise with observed images and inferred foundation mono-depths
Gaussians anchored on Motion Scaffolds
i
j
k
Time
Space
SCF node 1
SCF node 2
SCF node 3
SCF node 4
Fuse all GS from different time frames to the target rendering time (here time j)
Viz of Gaussian Trajectories: a Gaussian lifted from depth can be transported from any source time to any target time with underlying MoSca
(D) MoSca Photo-Fusion Stage Sec.3.2&3.4

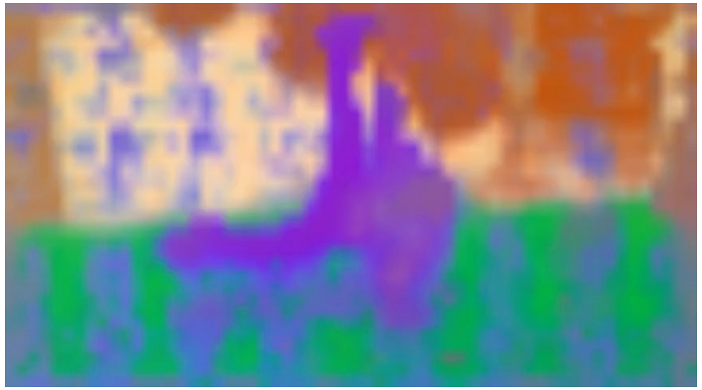


Long-term 2D Pixel Track

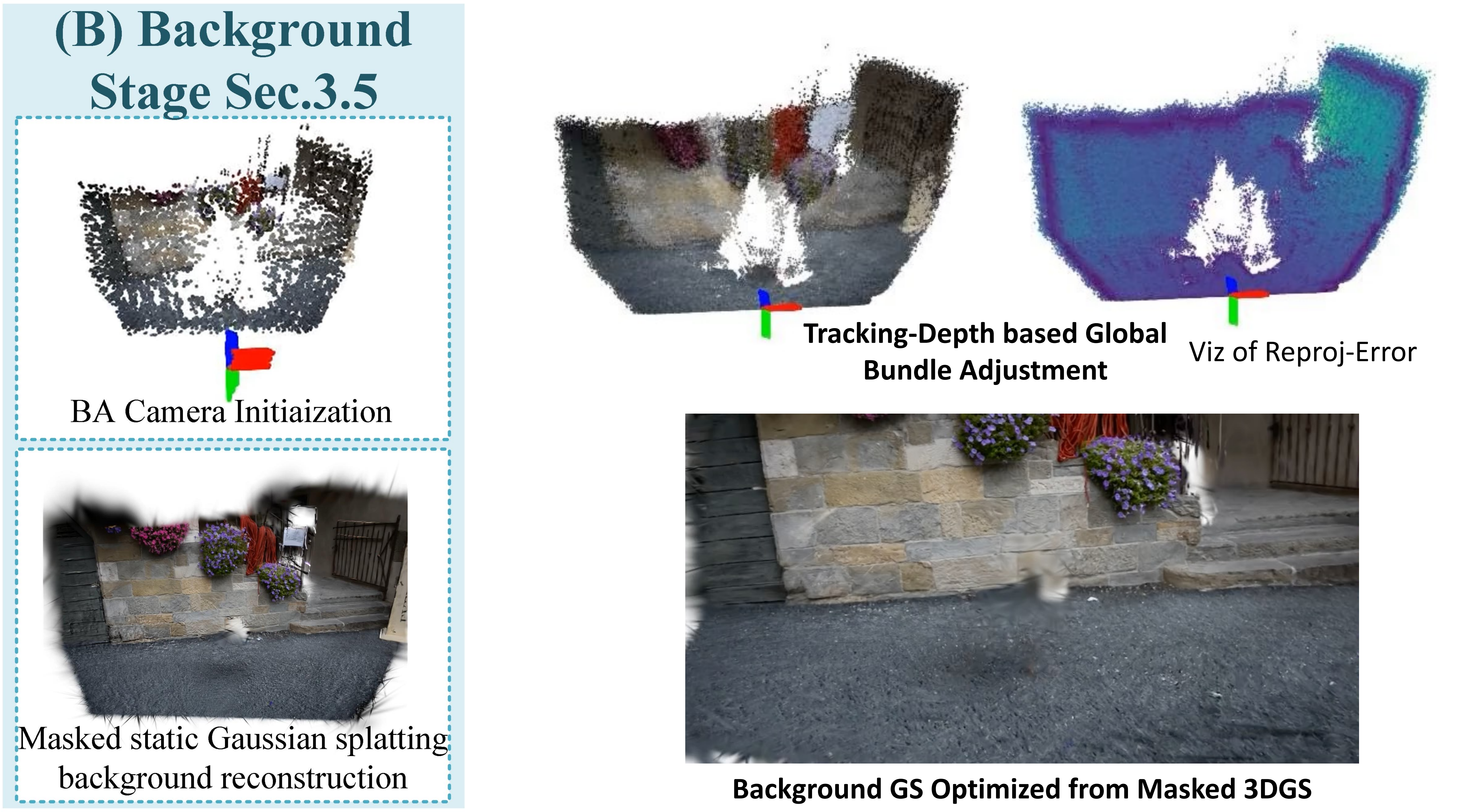
Monocular Metric Depth

Input

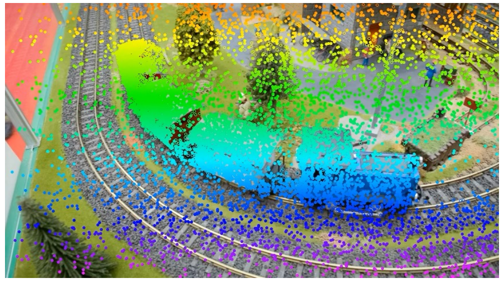
Epipolar Errors

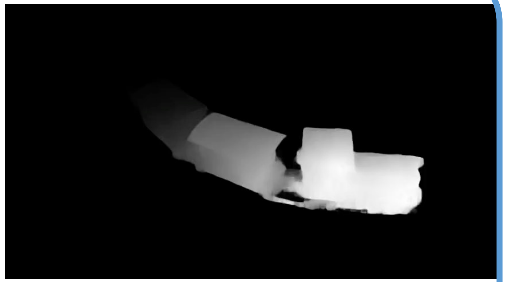


Semantic Features

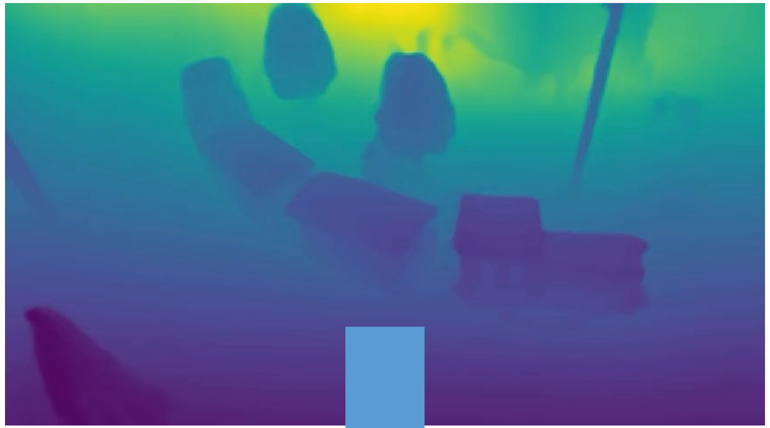

(B) Background
Stage Sec.3.5
BA Camera Initiaization
Masked static Gaussian splatting
background reconstruction
Tracking-Depth based Global
Bundle Adjustment
Viz of Reproj-Error
Background GS Optimized from Masked 3DGS

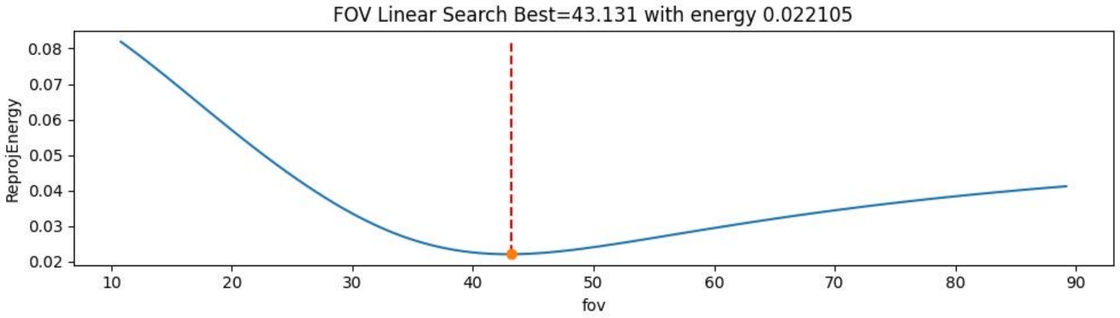
Background Geometry Initialization: Focal





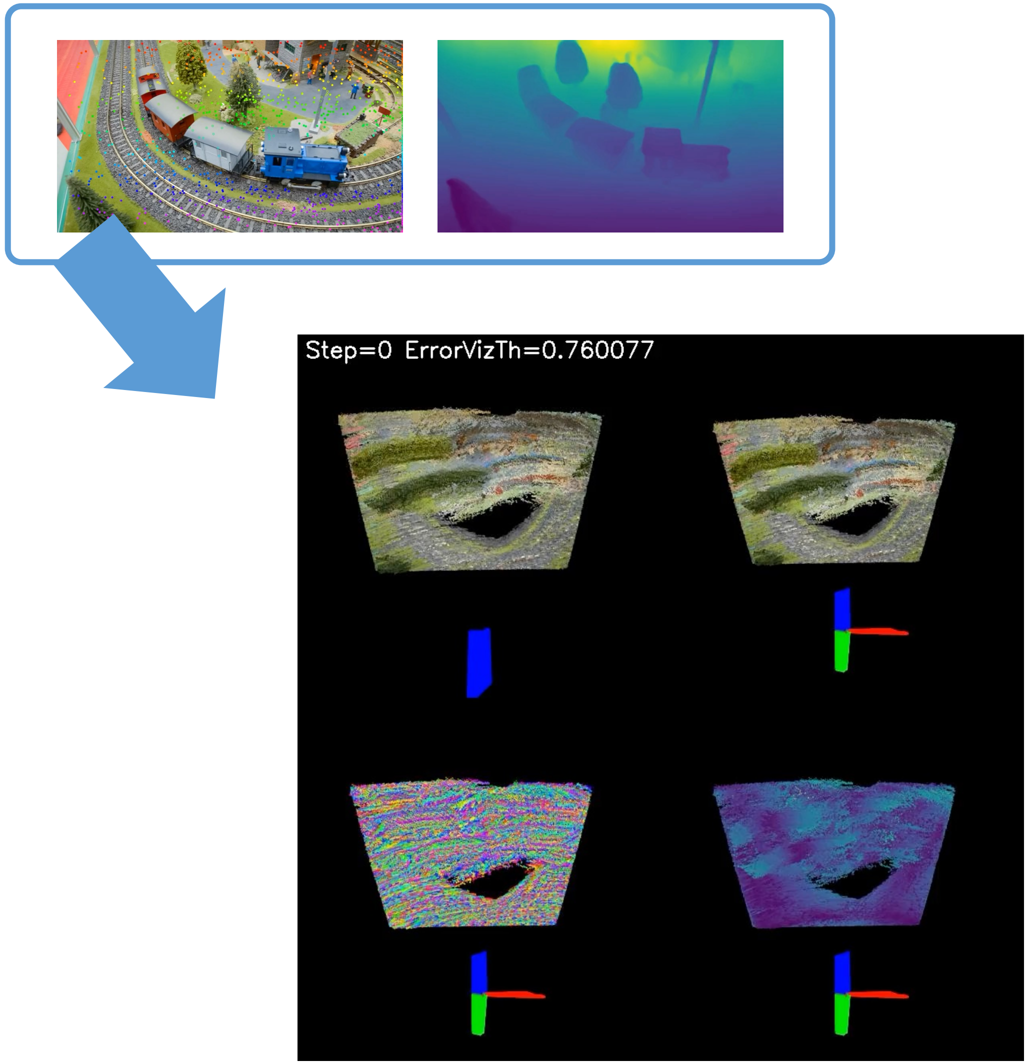
Tracks that always have small EpiError

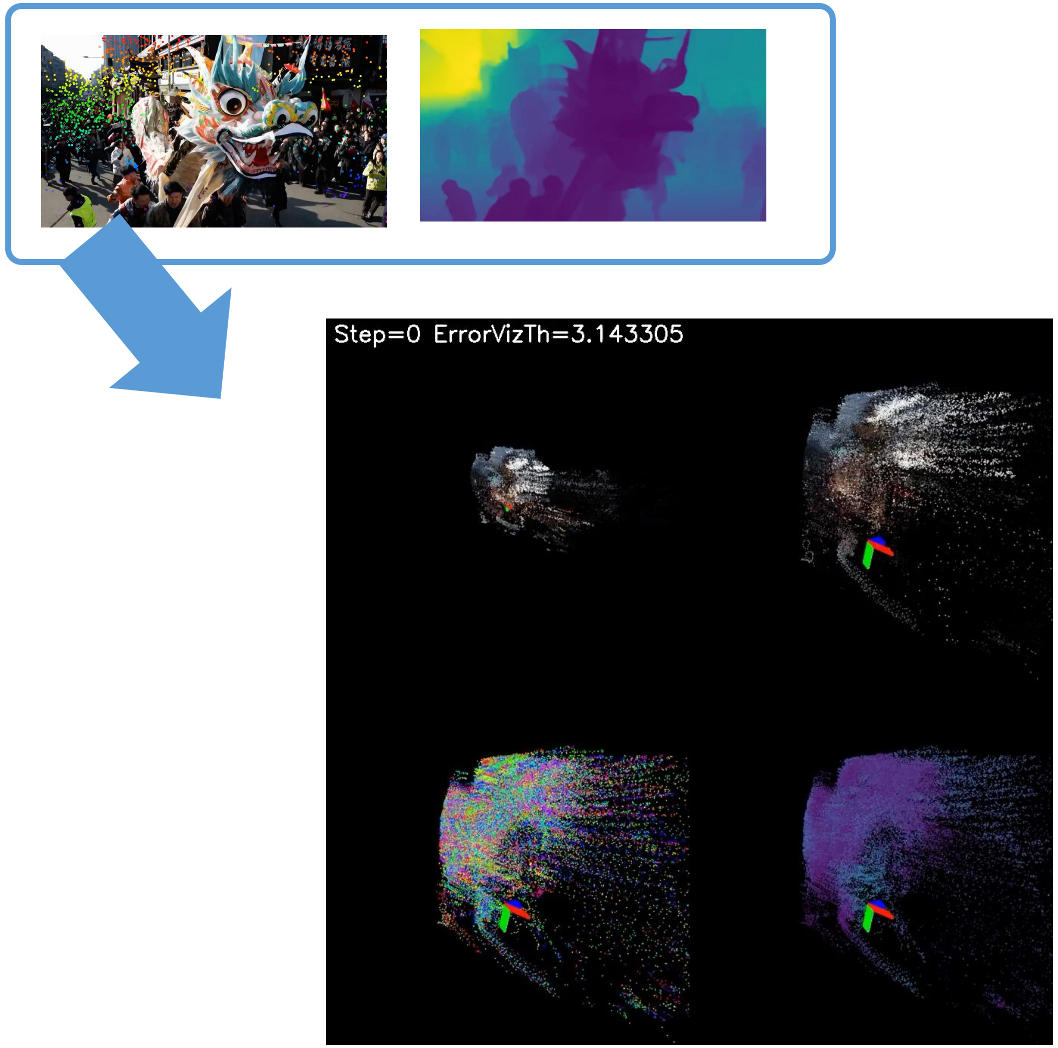


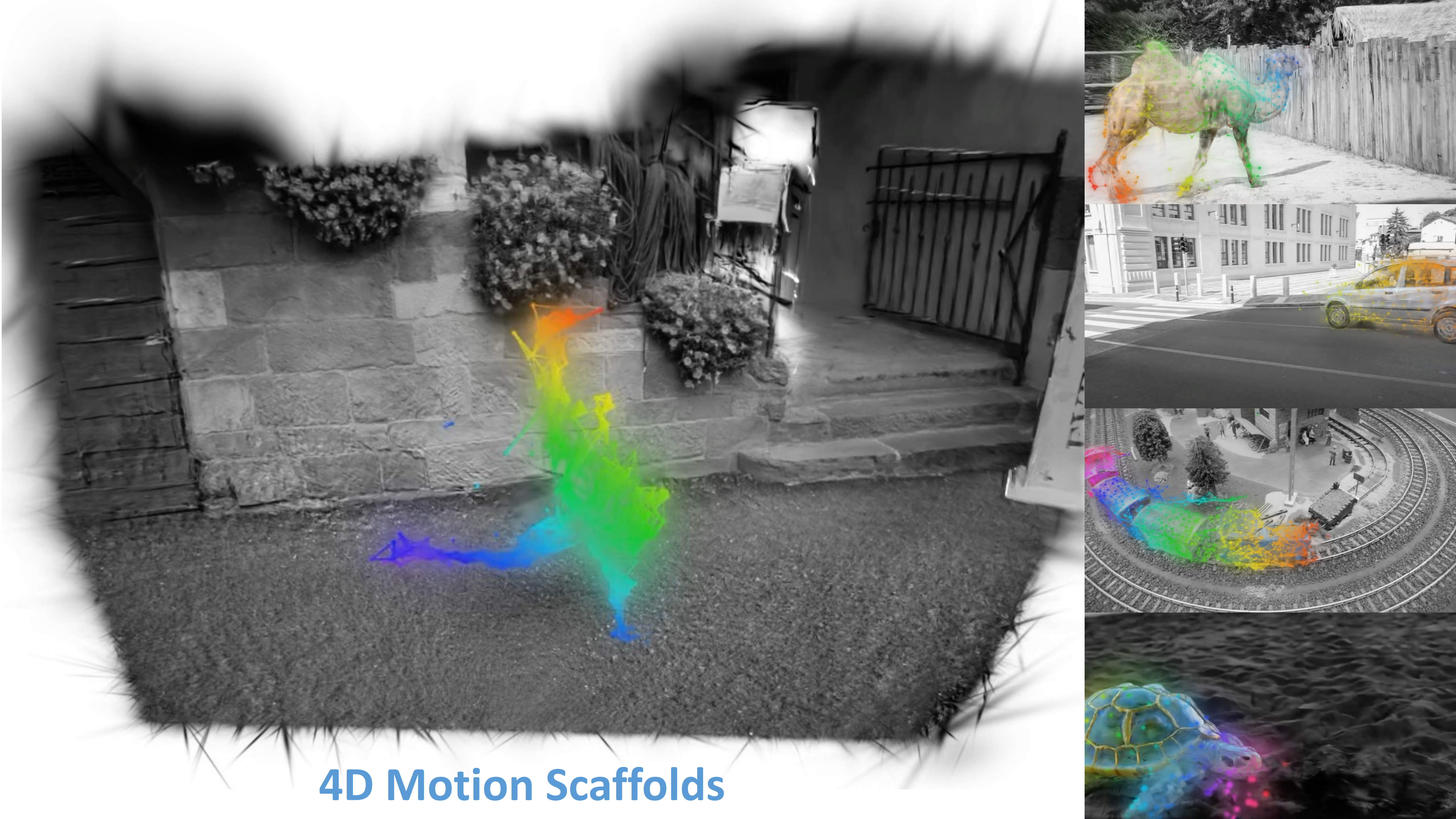


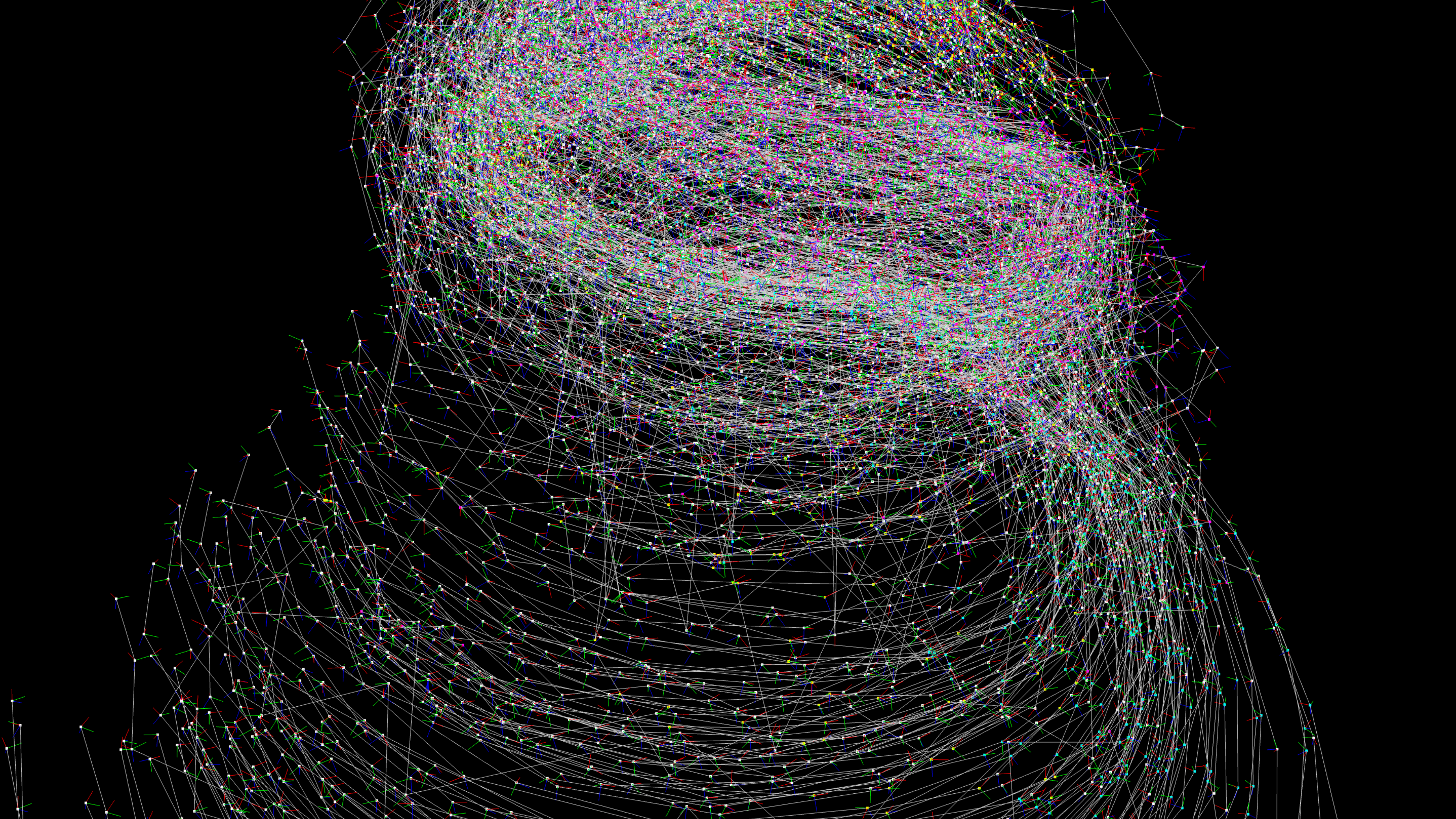
Enumerate FOV, Analytical Solve SIM(3) Procrustes,
Measure Re-projection error between all view pairs.

Background Geometry Initialization: BA

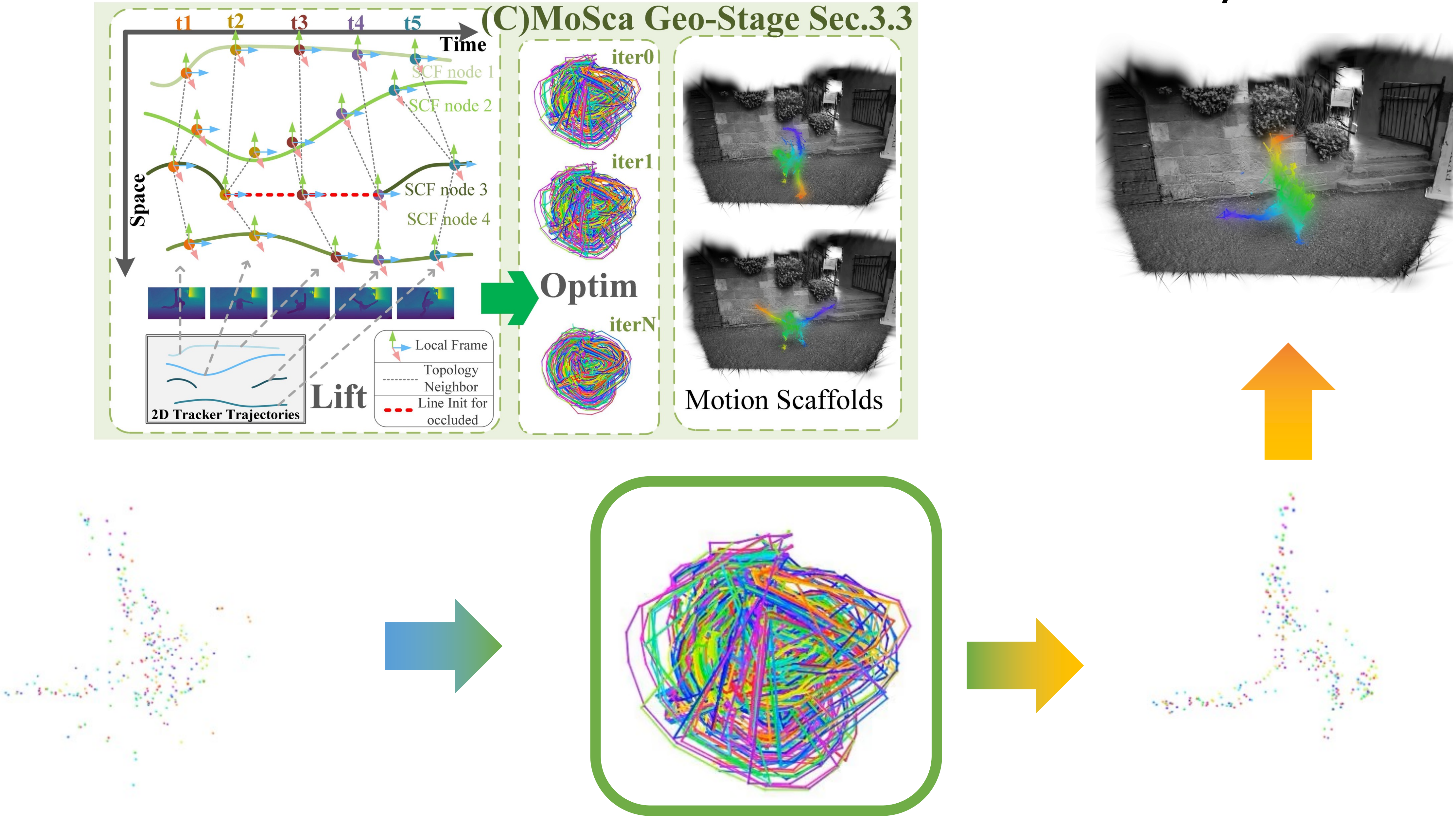





4D Motion Scaffolds



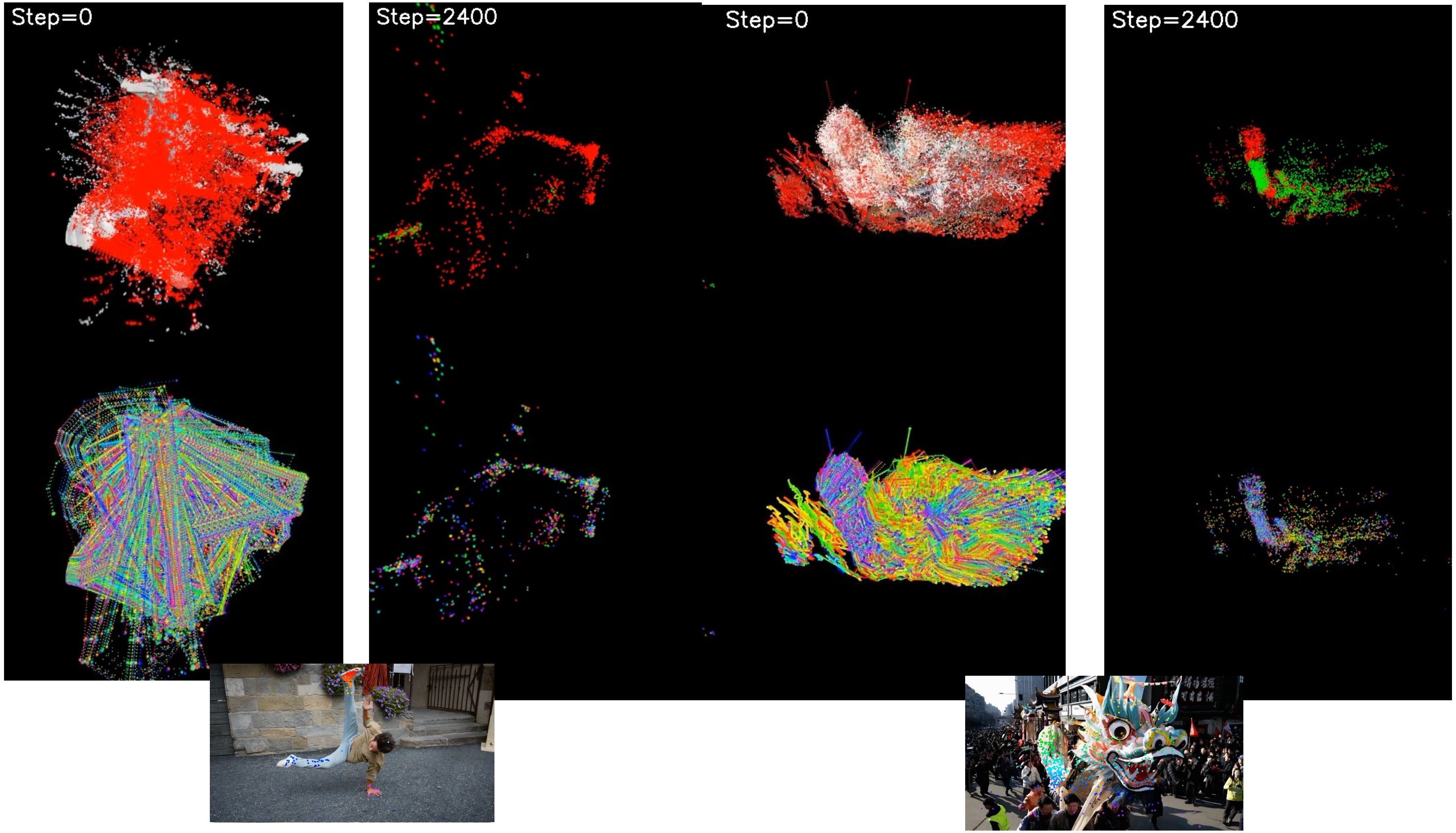
Ready for Fusion

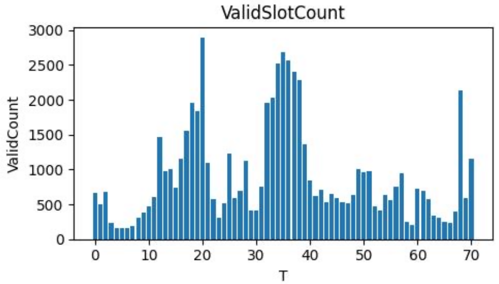


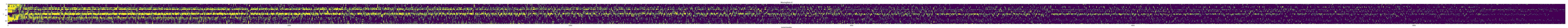
Lifted Initial Scaffolds **Geometric Optimization** **Optimized Scaffolds**

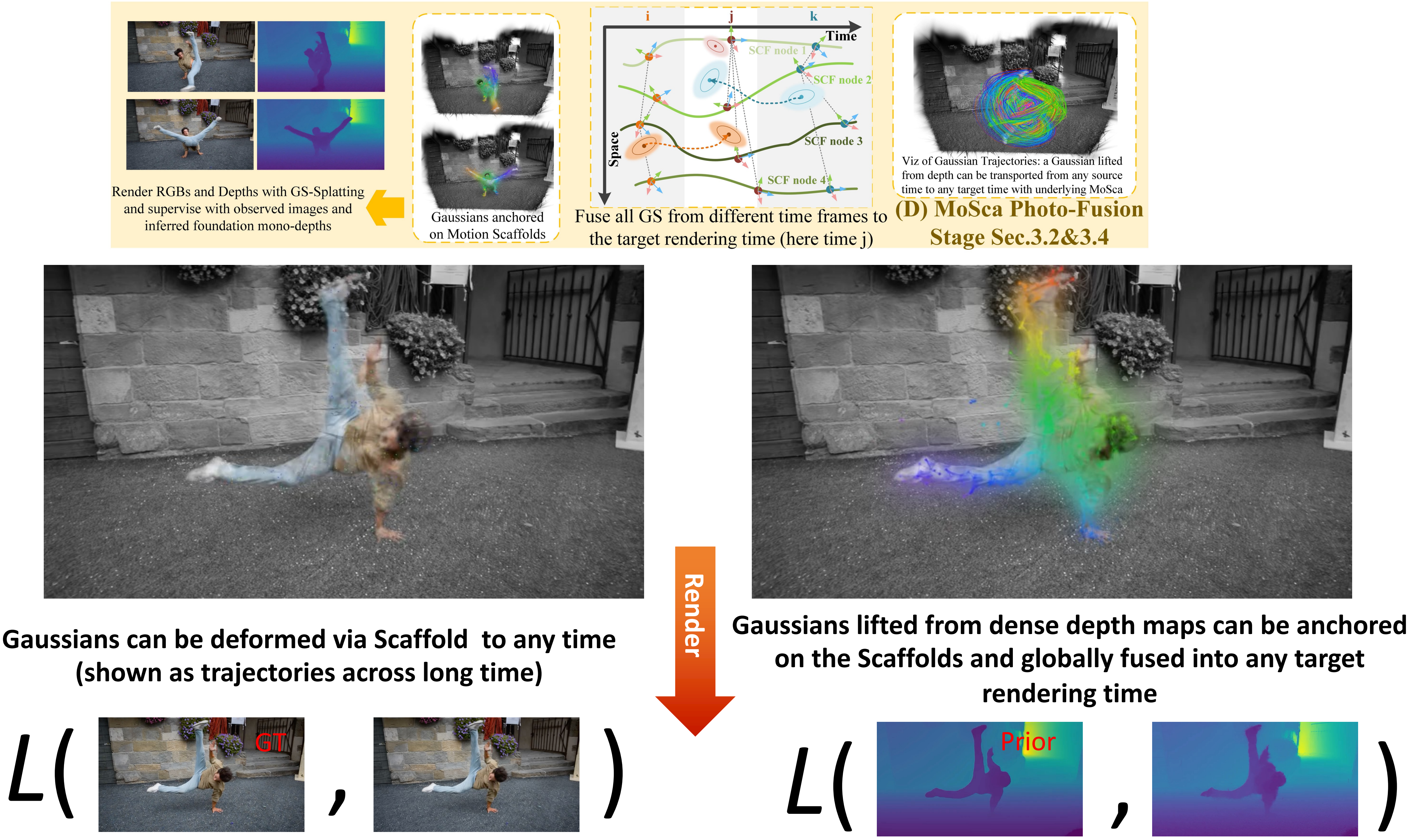
Dynamic 4D Scaffold

- Finally Optimize the unknown position and all node rotation with **ARAP** and **ACC** physical inspired energy



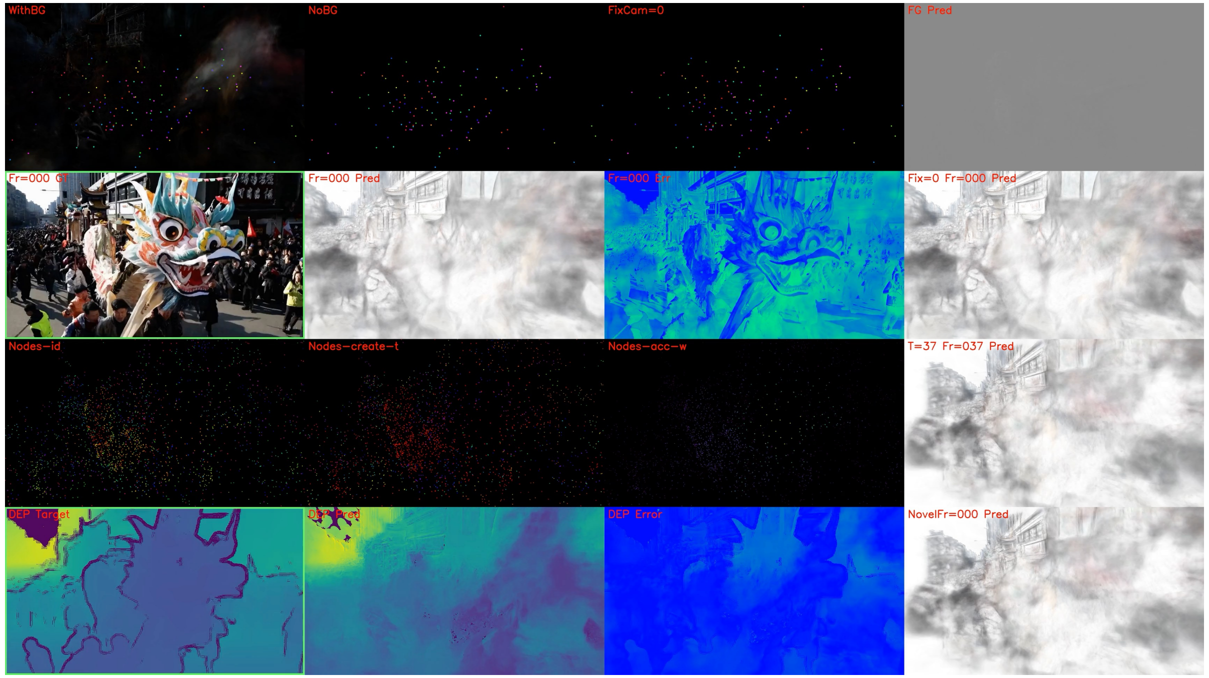




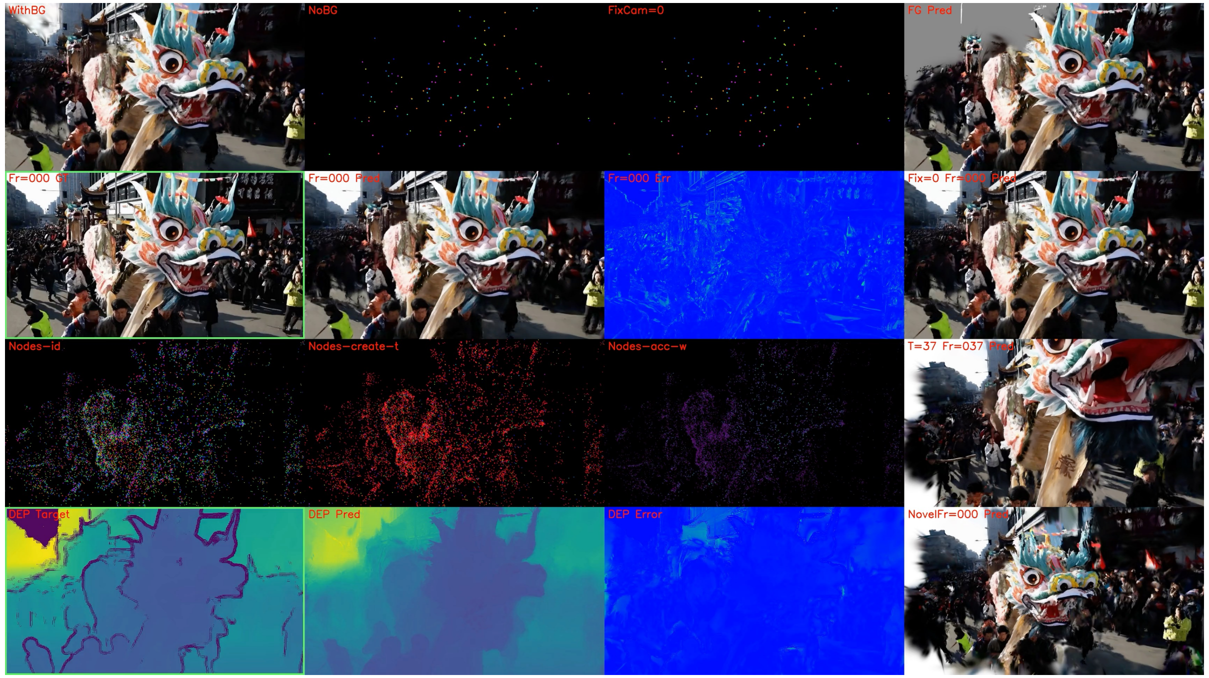


i
j
k
Time
Space
SCF node 1
SCF node 2
SCF node 3
SCF node 4
Render RGBs and Depths with GS-Splatting and supervise with observed images and inferred foundation mono-depths
Gaussians anchored on Motion Scaffolds
Fuse all GS from different time frames to the target rendering time (here time j)
Viz of Gaussian Trajectories: a Gaussian lifted from depth can be transported from any source time to any target time with underlying MoSca
(D) MoSca Photo-Fusion Stage Sec.3.2&3.4
Gaussians can be deformed via Scaffold to any time (shown as trajectories across long time)
Render
Gaussians lifted from dense depth maps can be anchored on the Scaffolds and globally fused into any target rendering time
L(GT ,)
L(Prior ,)

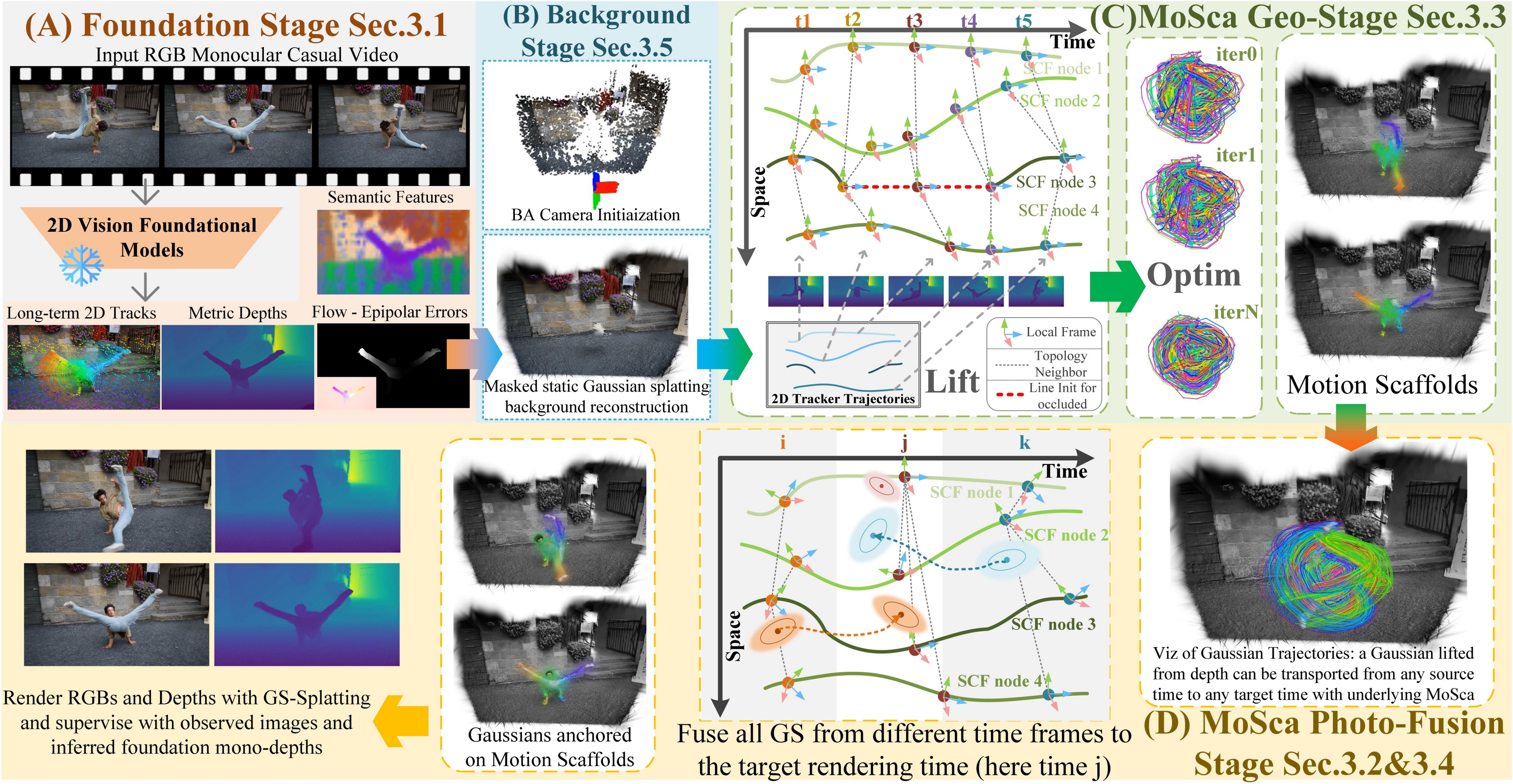
Photometric optimization



Step 0 Init



Optimized

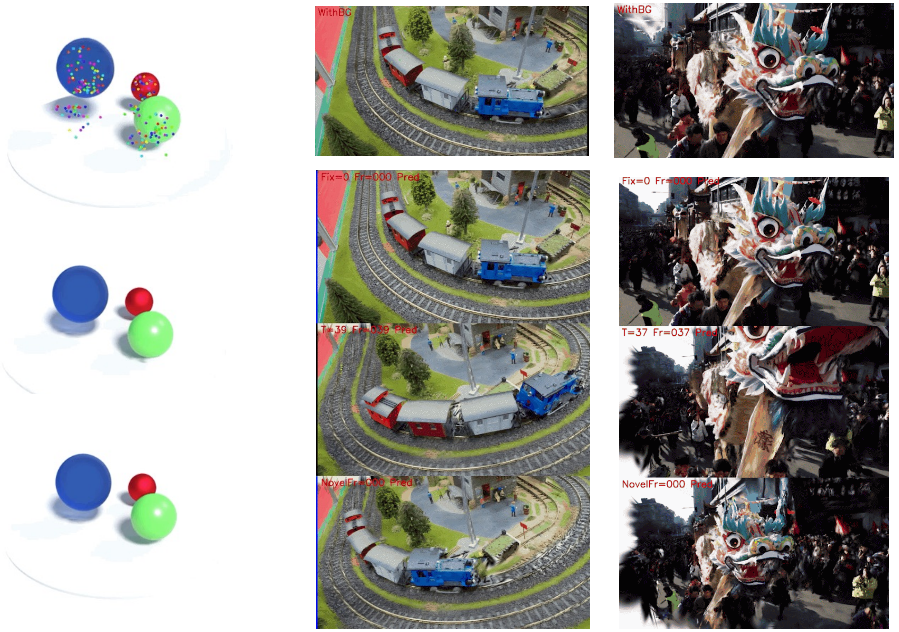


(A) Foundation Stage Sec.3.1
Input RGB Monocular Casual Video
2D Vision Foundational Models
Semantic Features
Long-term 2D Tracks
Metric Depths
Flow - Epipolar Errors
(B) Background Stage Sec.3.5
BA Camera Initiaization
Masked static Gaussian splatting background reconstruction
(C)MoSca Geo-Stage Sec.3.3
t1
t2
t3
t4
t5
Time
Space
SCF node 1
SCF node 2
SCF node 3
SCF node 4
2D Tracker Trajectories
Lift
Local Frame
Topology Neighbor
Line Init for occluded
iter0
iter1
iterN
Optim
Motion Scaffolds
Render RGBs and Depths with GS-Splatting and supervise with observed images and inferred foundation mono-depths
Gaussians anchored on Motion Scaffolds
i
j
k
Fuse all GS from different time frames to the target rendering time (here time j)
Viz of Gaussian Trajectories: a Gaussian lifted from depth can be transported from any source time to any target time with underlying MoSca
(D) MoSca Photo-Fusion Stage Sec.3.2&3.4

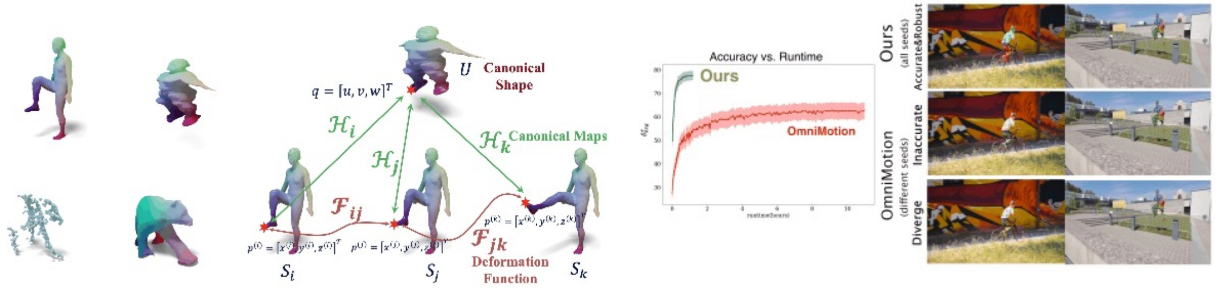
Overview of today’s talk

NAP: Neural Articulation Prior

GART: Gaussian Articulated Template Models



DynMF and MoSca



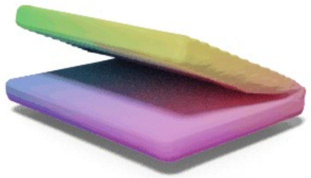
CaDeX and CaDeX++

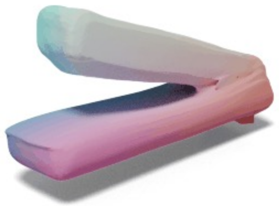
CaDeX: Learning Canonical Deformation Coordinate Space for Dynamic Surface Representation via Neural Homeomorphism

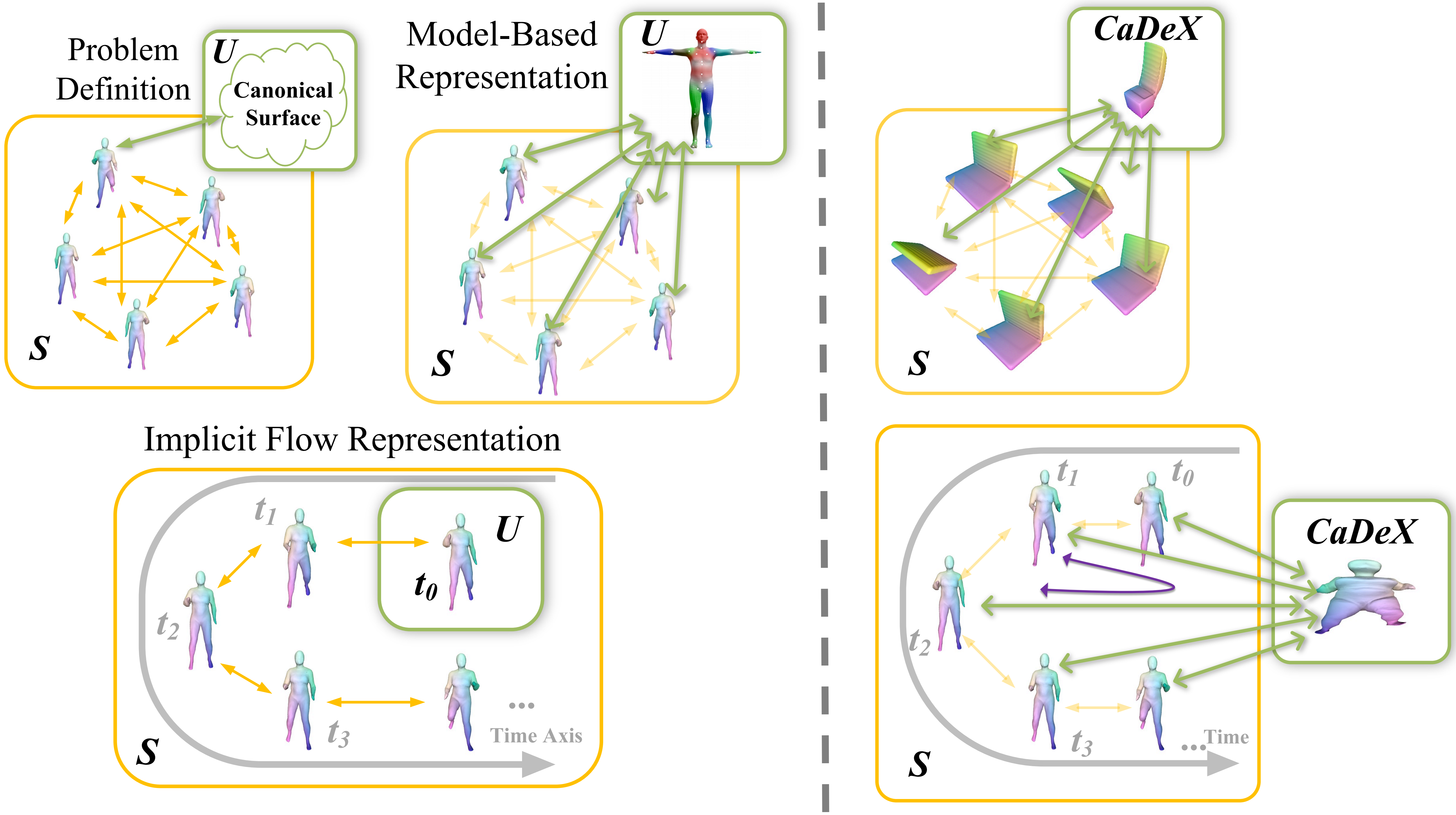
Jiahui Lei   Kostas Daniilidis

University of Pennsylvania

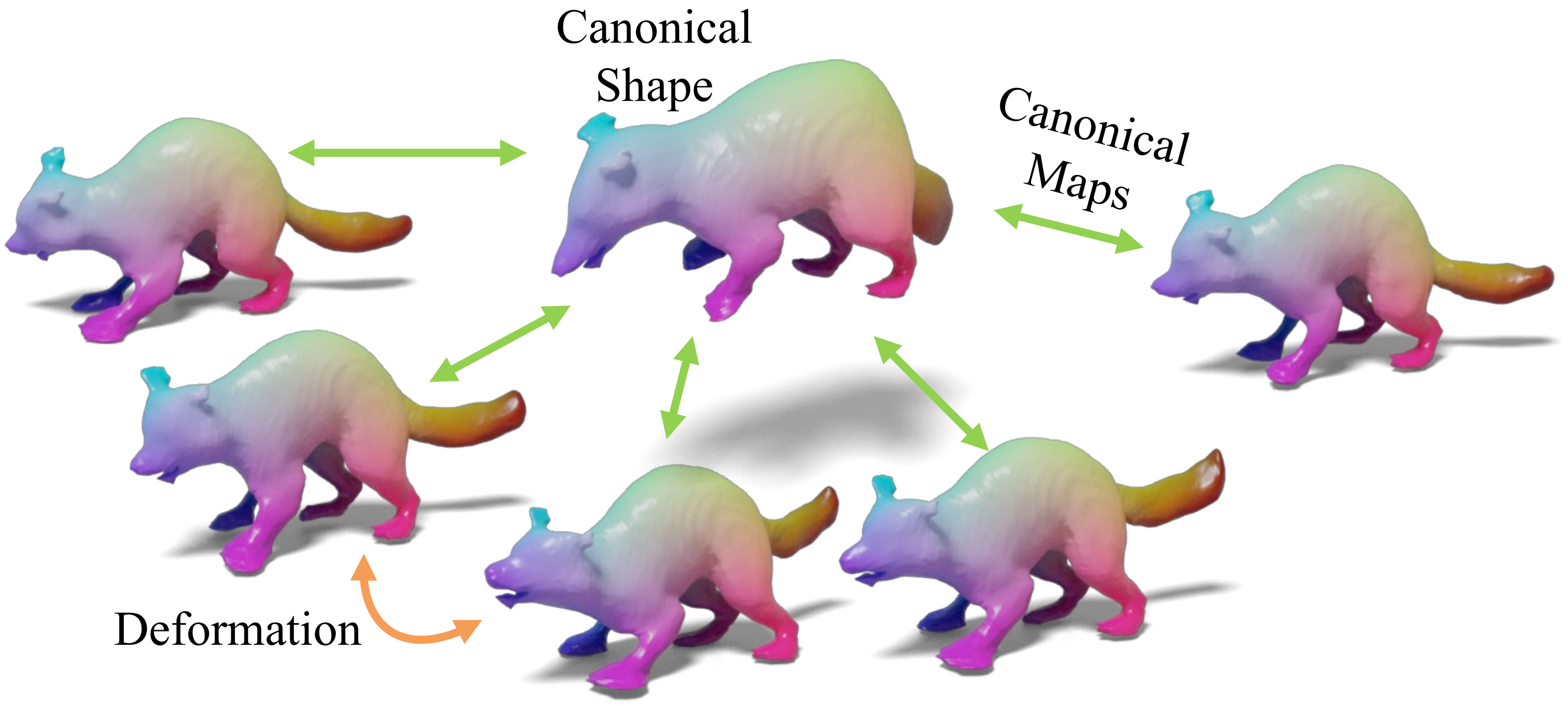
CVPR 2022 [**Part I Method** (with narration)]



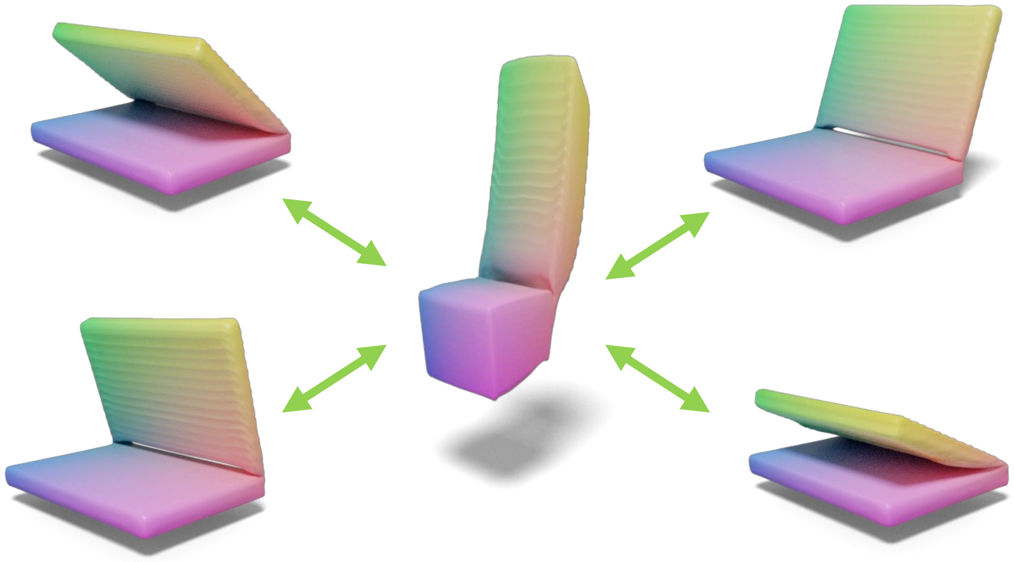




Problem Definition
U
Canonical Surface
S
Model-Based Representation
U
S
CaDeX
S
Implicit Flow Representation
t_1
U
t_0
t_2
t_3
...
Time Axis
S
t_1
t_0
CaDeX
t_2
t_3
...Time
S



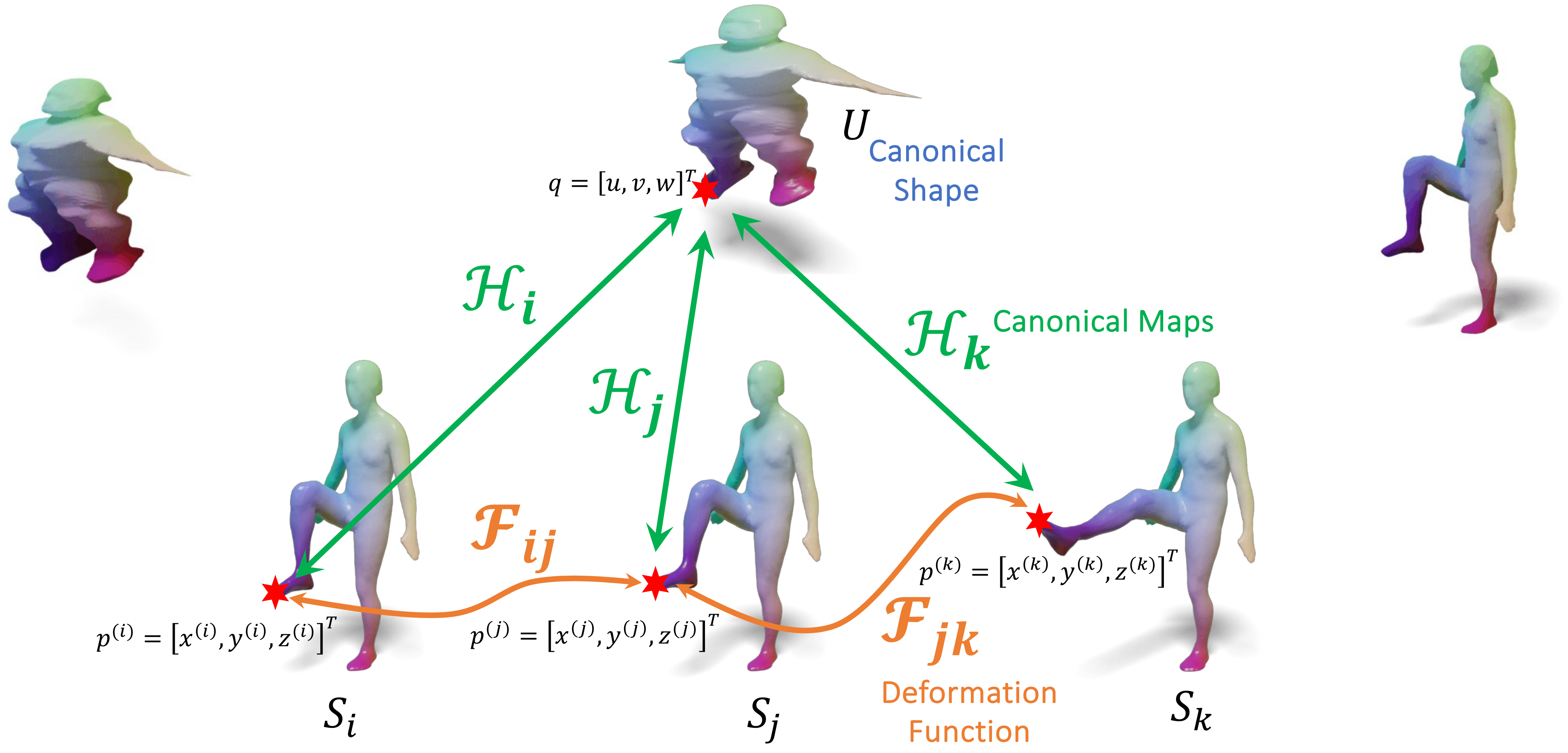
Representation



Deformation Factorization: $p^{(j)} = \mathcal{F}_{ij}(p^{(i)}) = \mathcal{H}_j^{-1} \circ \mathcal{H}_i(p^{(i)})$

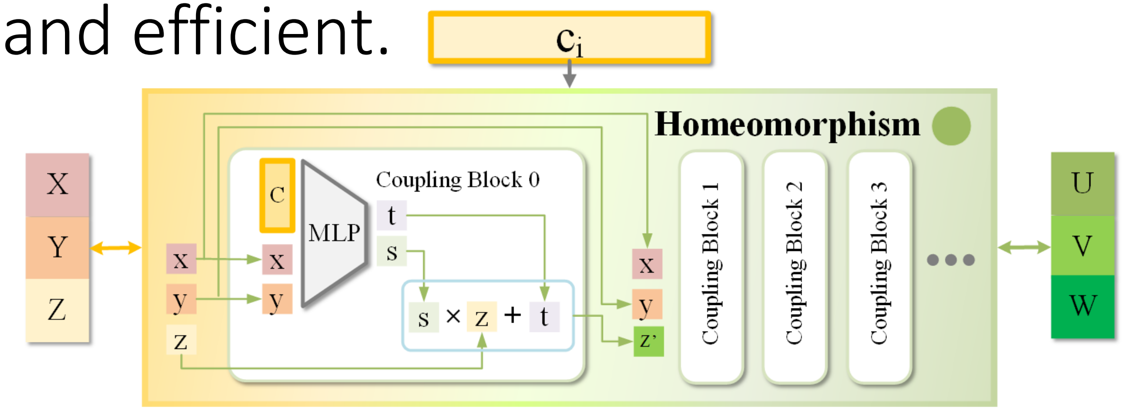
Canonical Shape: $U = \{q \mid q = [u, v, w]^T, OccField(q) = level\}$

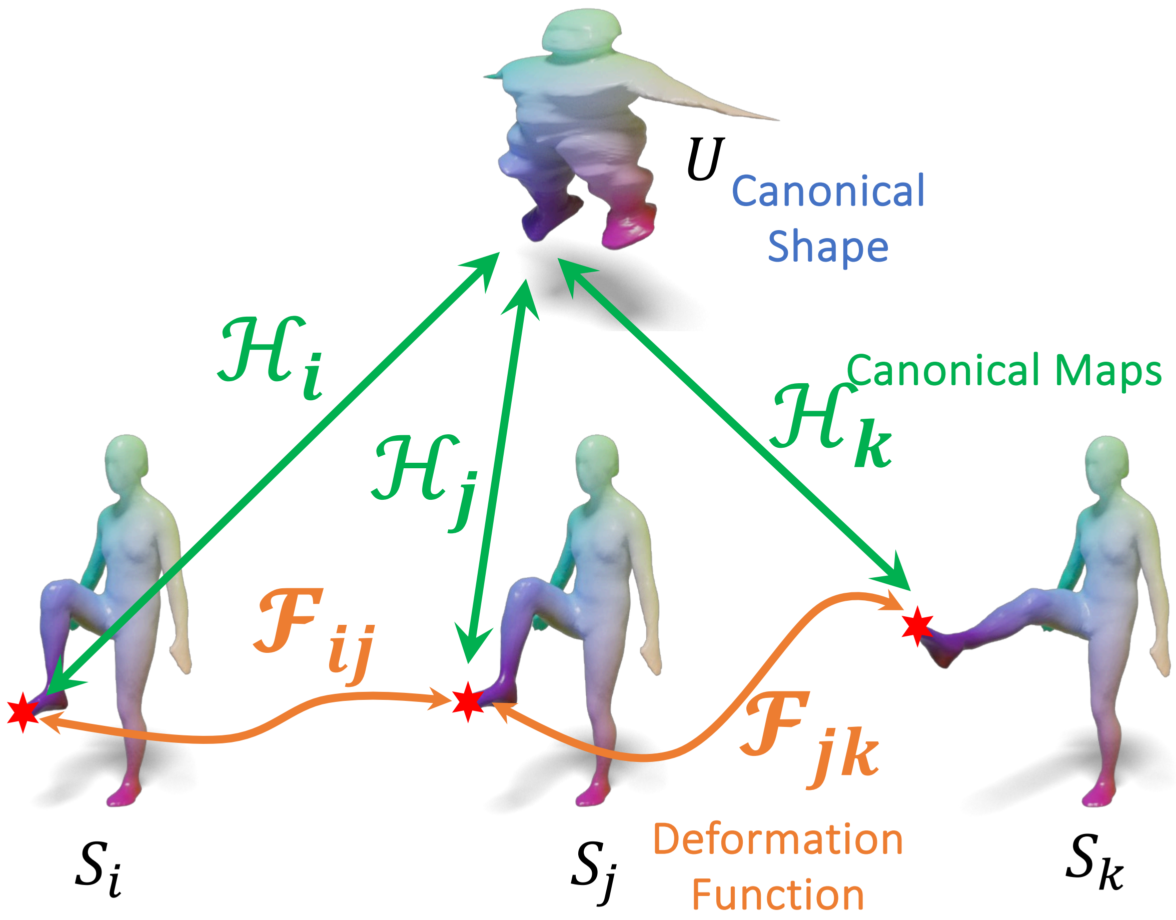
Deformed Shapes: $S_i = \{p \mid p = [x^{(i)}, y^{(i)}, z^{(i)}]^T = \mathcal{H}_i^{-1}(q), \forall q \in U\}$



Deformation Factorization: $p^{(j)} = \mathcal{F}_{ij}(p^{(i)}) = \mathcal{H}_j^{-1} \circ \mathcal{H}_i(p^{(i)})$
Canonical Shape: $U = \{q \mid q = [u, v, w]^T, OccField(q) = level\}$
Deformed Shapes: $S_i = \{p \mid p = [x^{(i)}, y^{(i)}, z^{(i)}]^T = \mathcal{H}_i^{-1}(q), \forall q \in U\}$

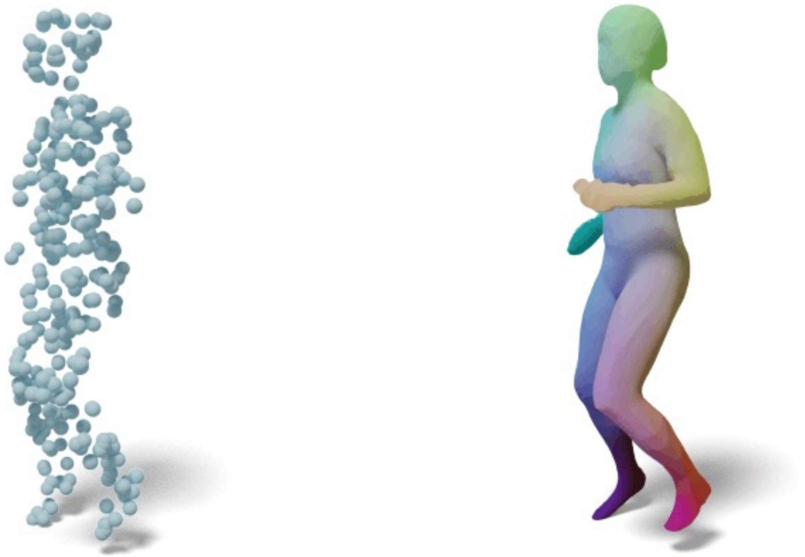
$\mathcal{H}_i$ is implemented by conditional Real-NVP or NICE that are simple and efficient.

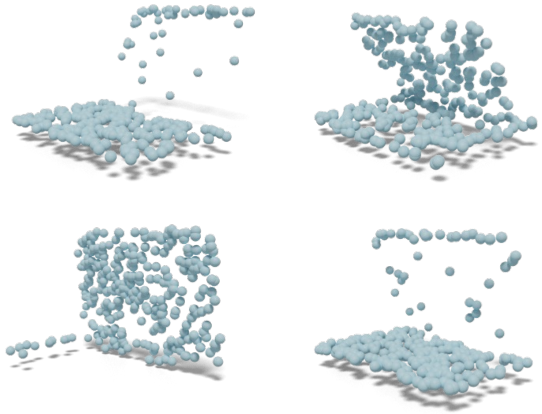


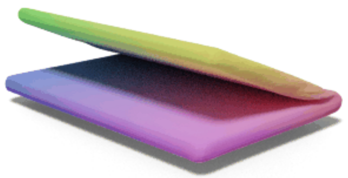


The deformation/correspondence factorization and its implementation guarantees:

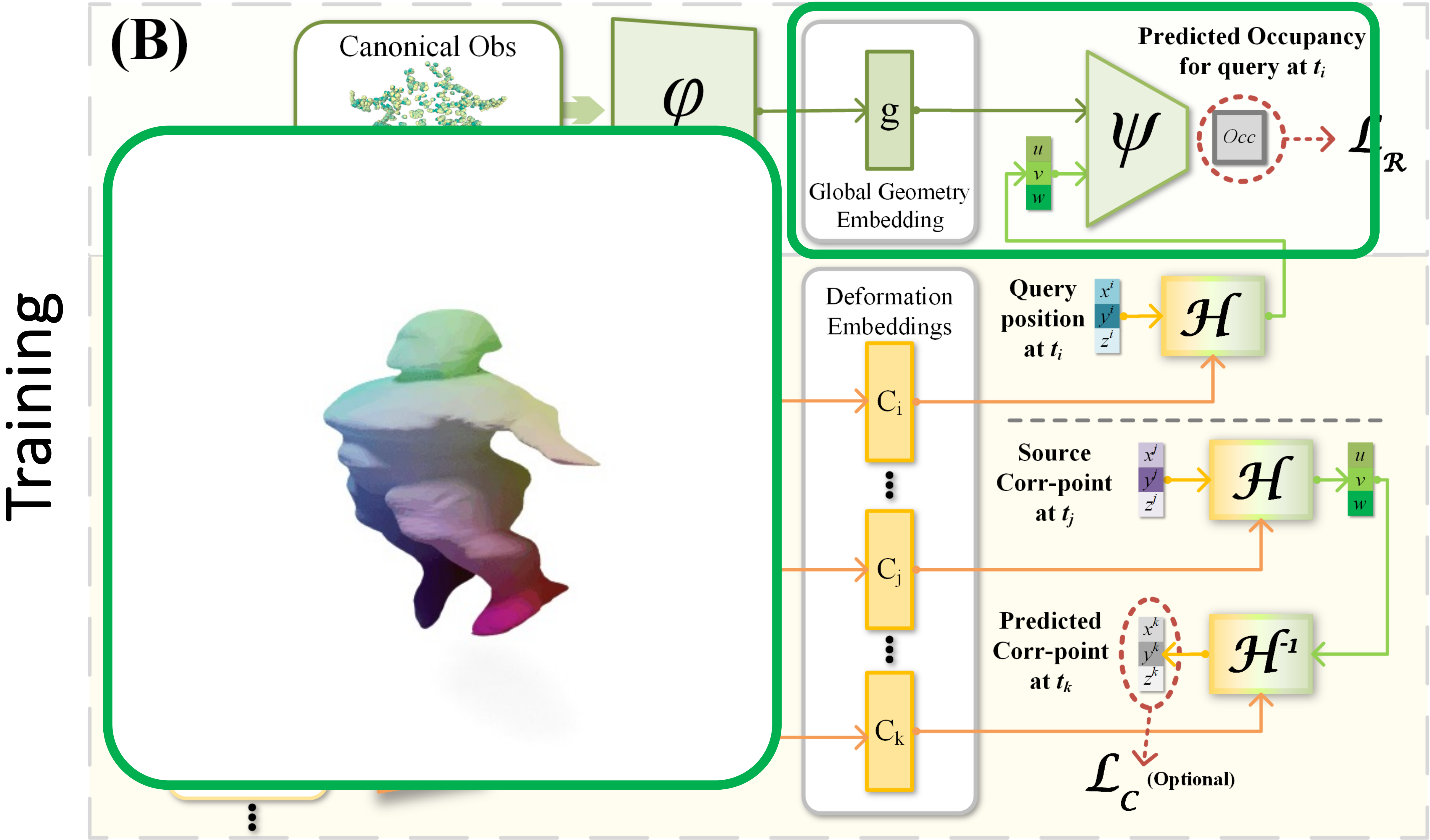
- Cycle consistency
- Topology Preservation
- Volume Conservation (Optional, if use NICE)



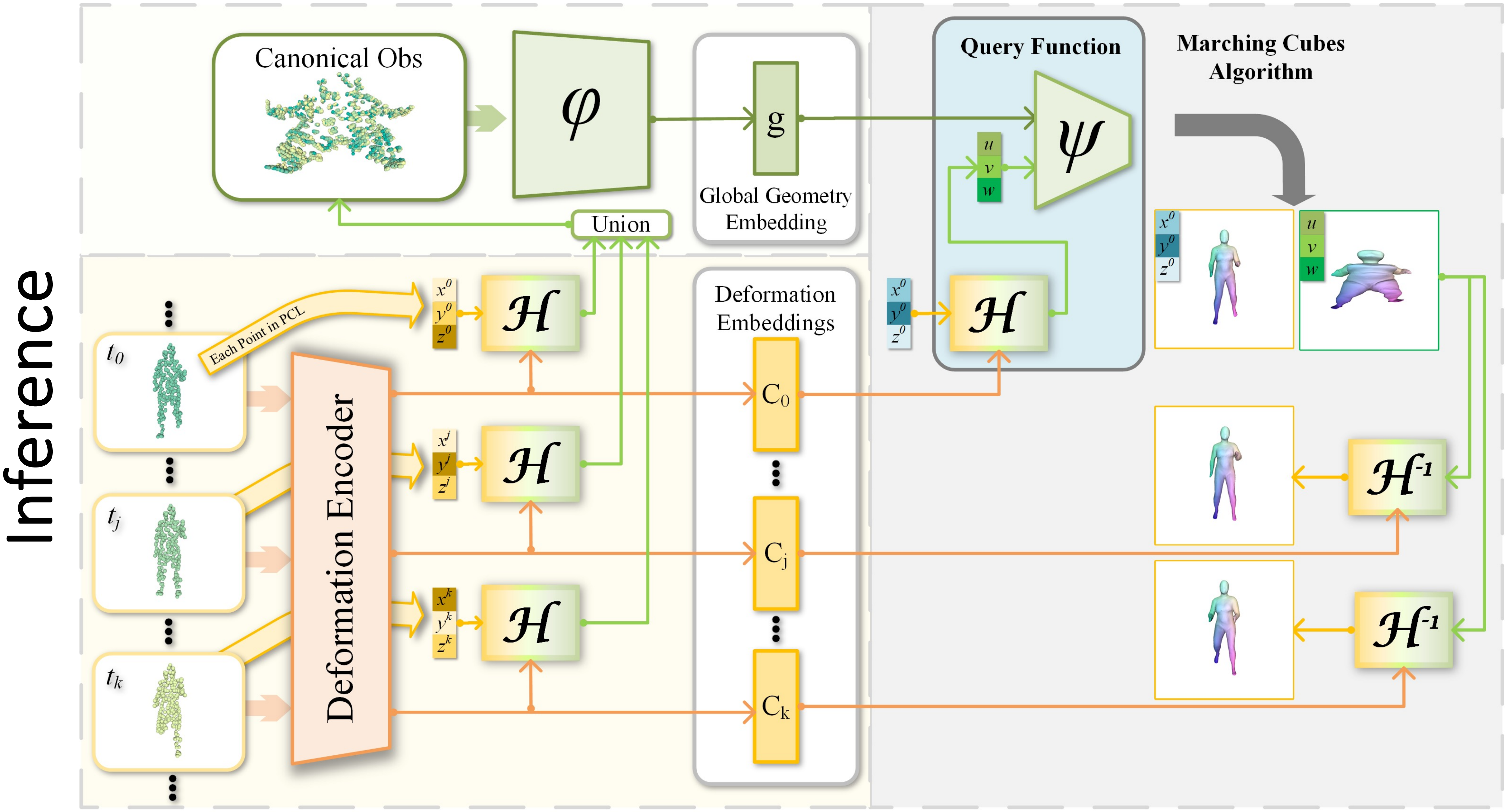




Architecture

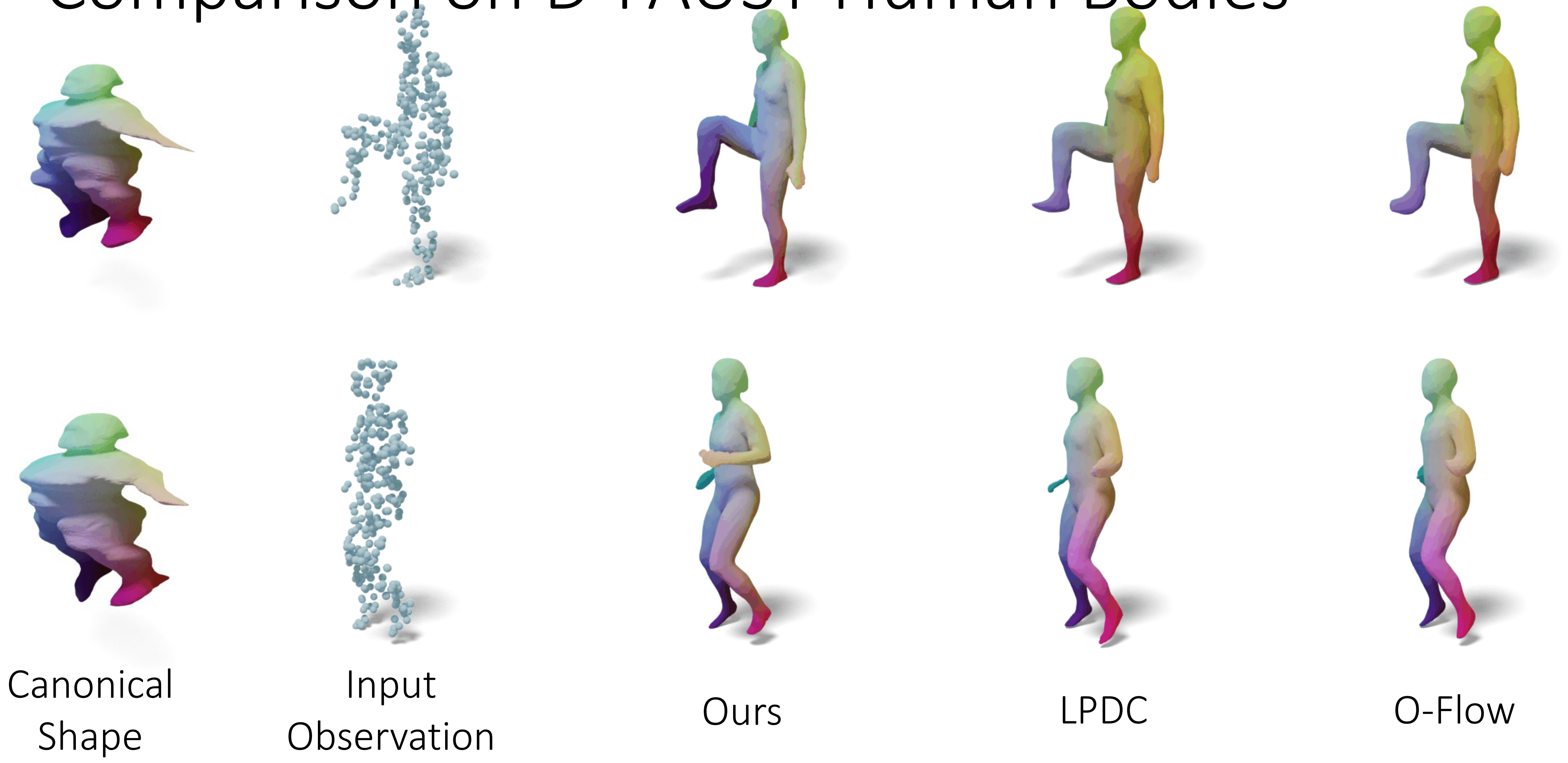


Training
(B)
Canonical Obs
φ
g
Global Geometry Embedding
Predicted Occupancy for query at t_i
ψ
u
v
w
Occ
$\mathcal{L}_R$
Deformation Embeddings
C_i
C_j
C_k
Query position at t_i
x^i
y^i
z^i
$\mathcal{H}$
Source Corr-point at t_j
x^j
y^j
z^j
$\mathcal{H}$
u
v
w
Predicted Corr-point at t_k
x^k
y^k
z^k
$\mathcal{H}^{-1}$
$\mathcal{L}_C$ (Optional)

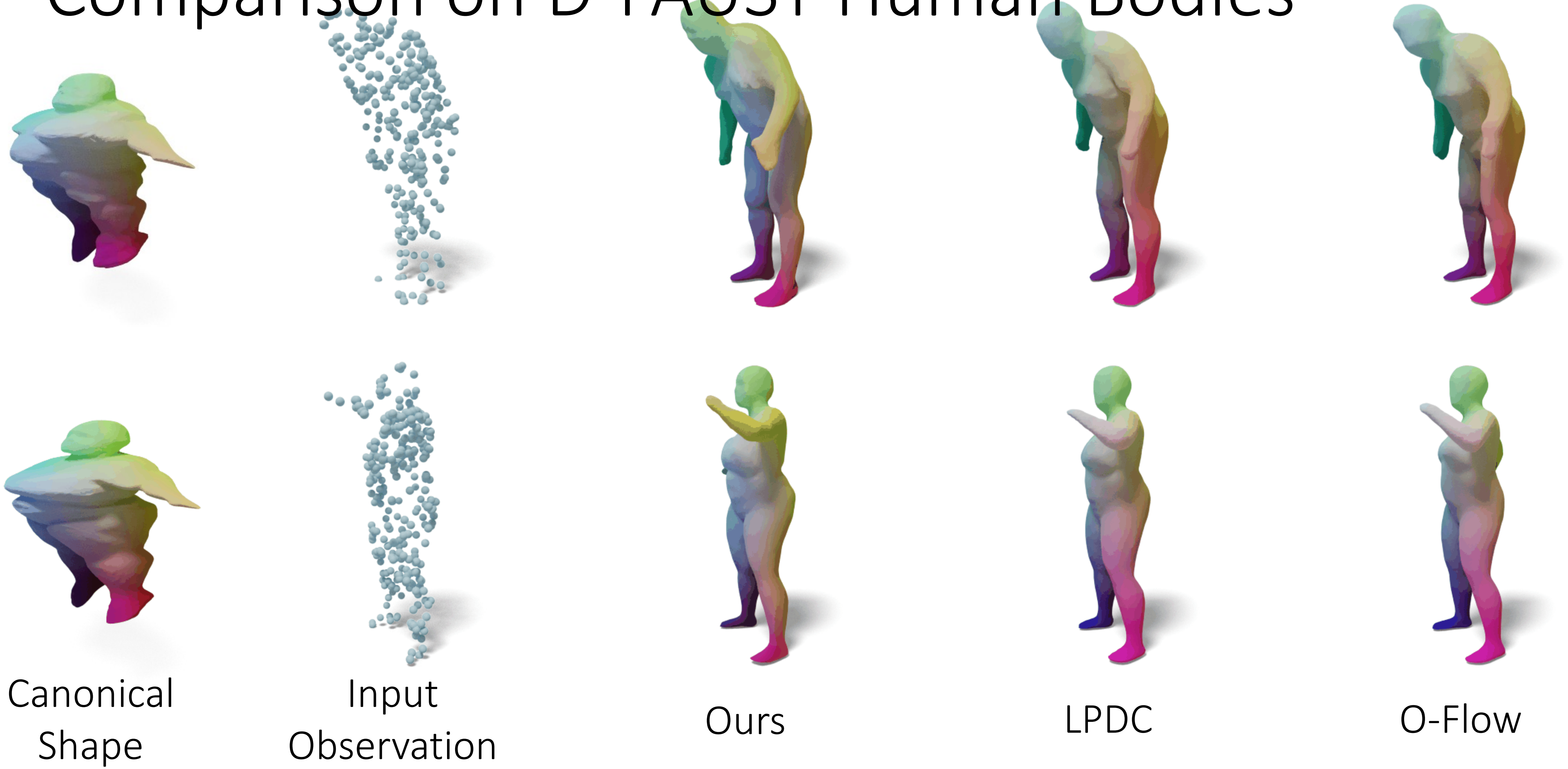


Inference
Canonical Obs
φ
g
Global Geometry Embedding
Query Function
u
v
w
ψ
Marching Cubes Algorithm
Union
Each Point in PCL
t_0
t_j
t_k
x^0 y^0 z^0
x^j y^j z^j
x^k y^k z^k
$\mathcal{H}$
$\mathcal{H}^{-1}$
Deformation Encoder
Deformation Embeddings
C_0
C_j
C_k

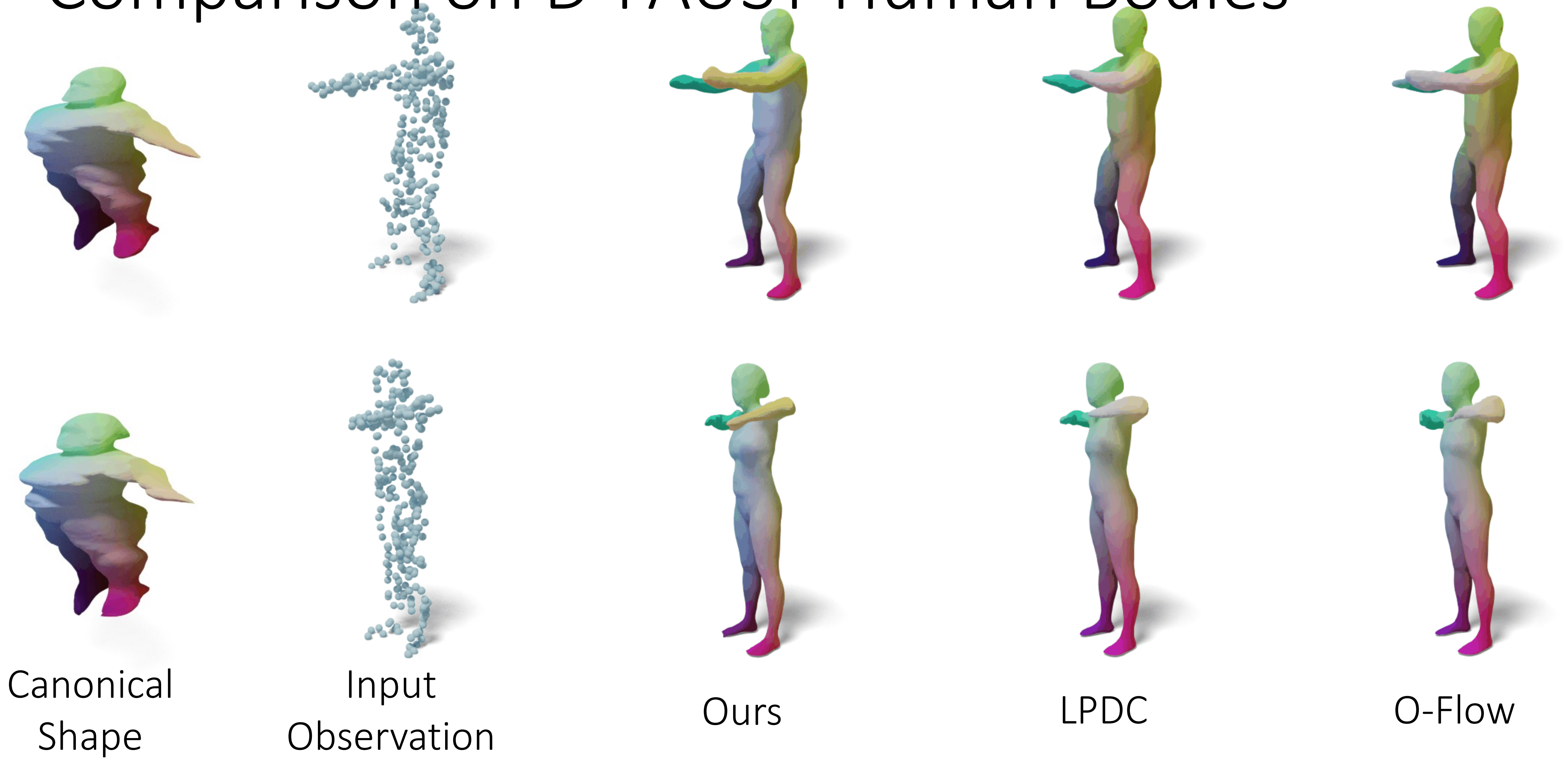
Comparison on D-FAUST Human Bodies



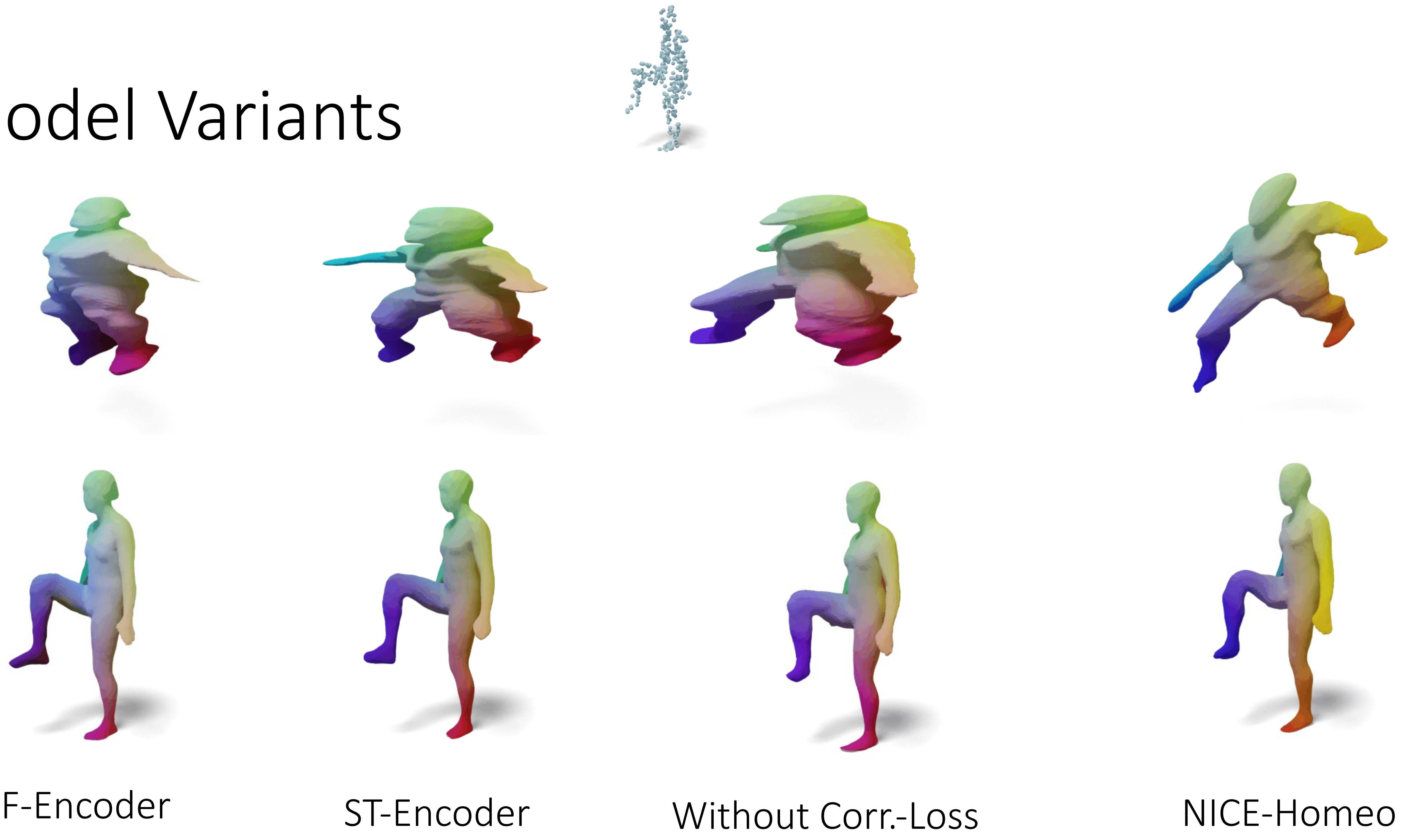
Comparison on D-FAUST Human Bodies



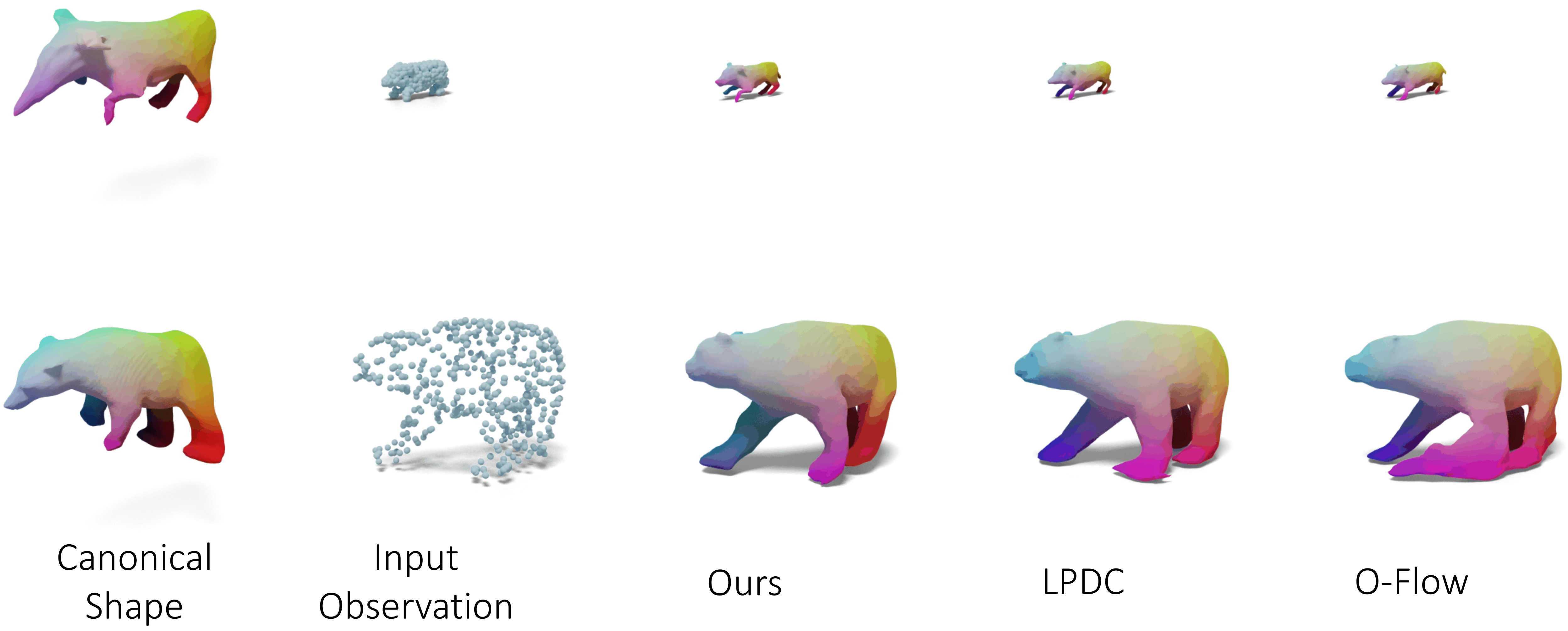
Comparison on D-FAUST Human Bodies



Model Variants

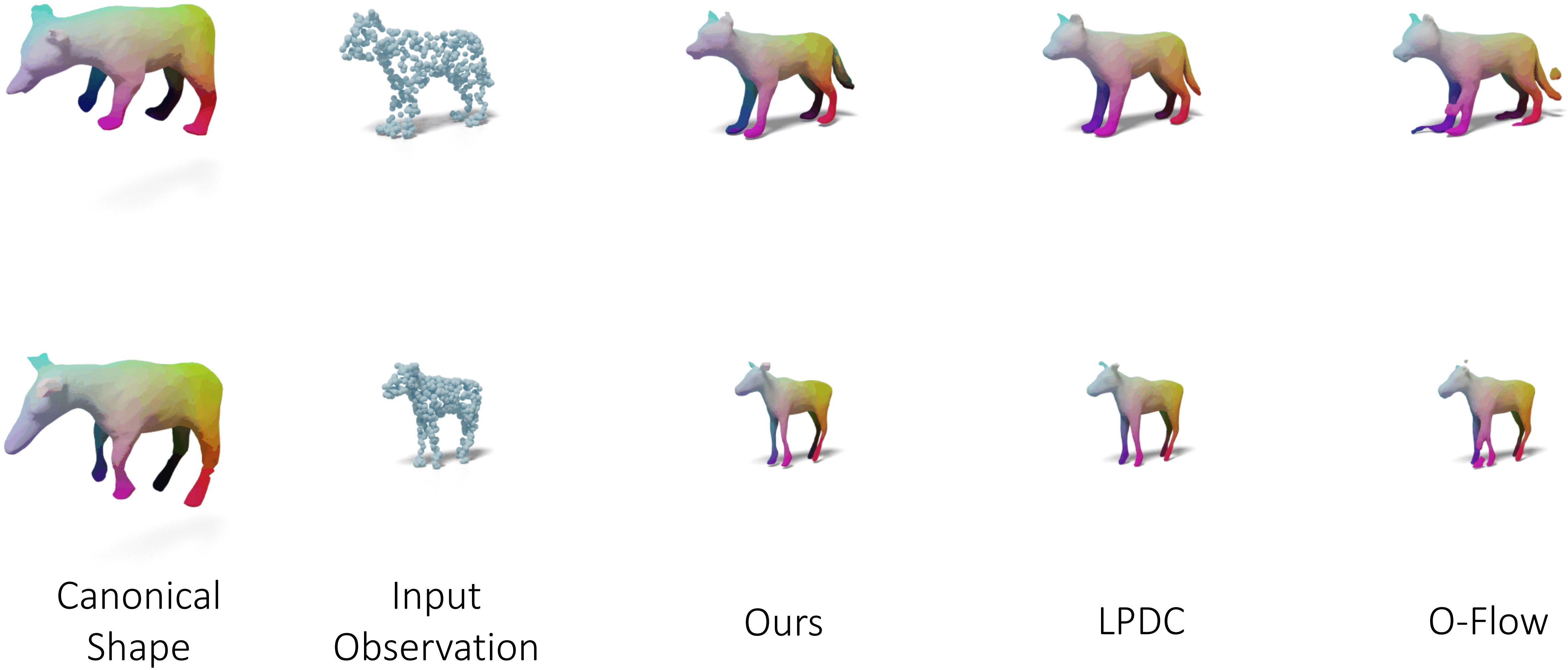


Comparison on DeformingThings4D Animals

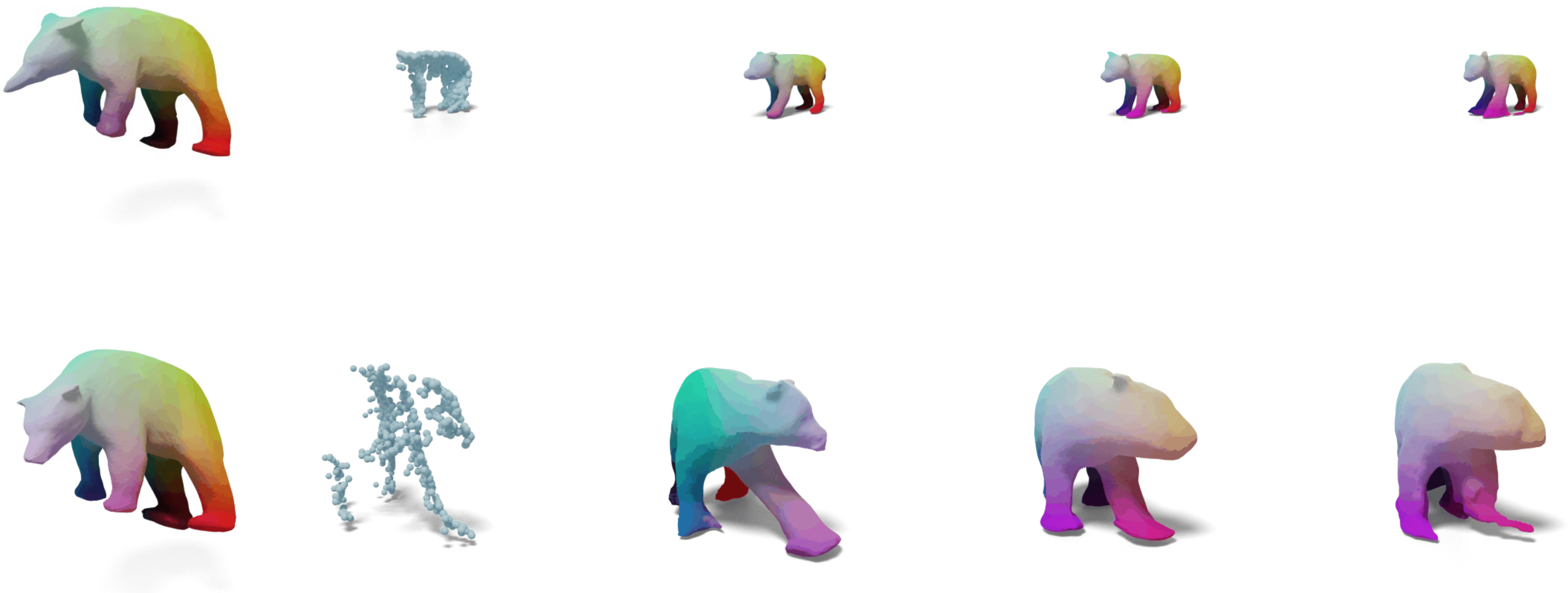


Comparison on DeformingThings4D Animals

Comparison on DeformingThings4D Animals

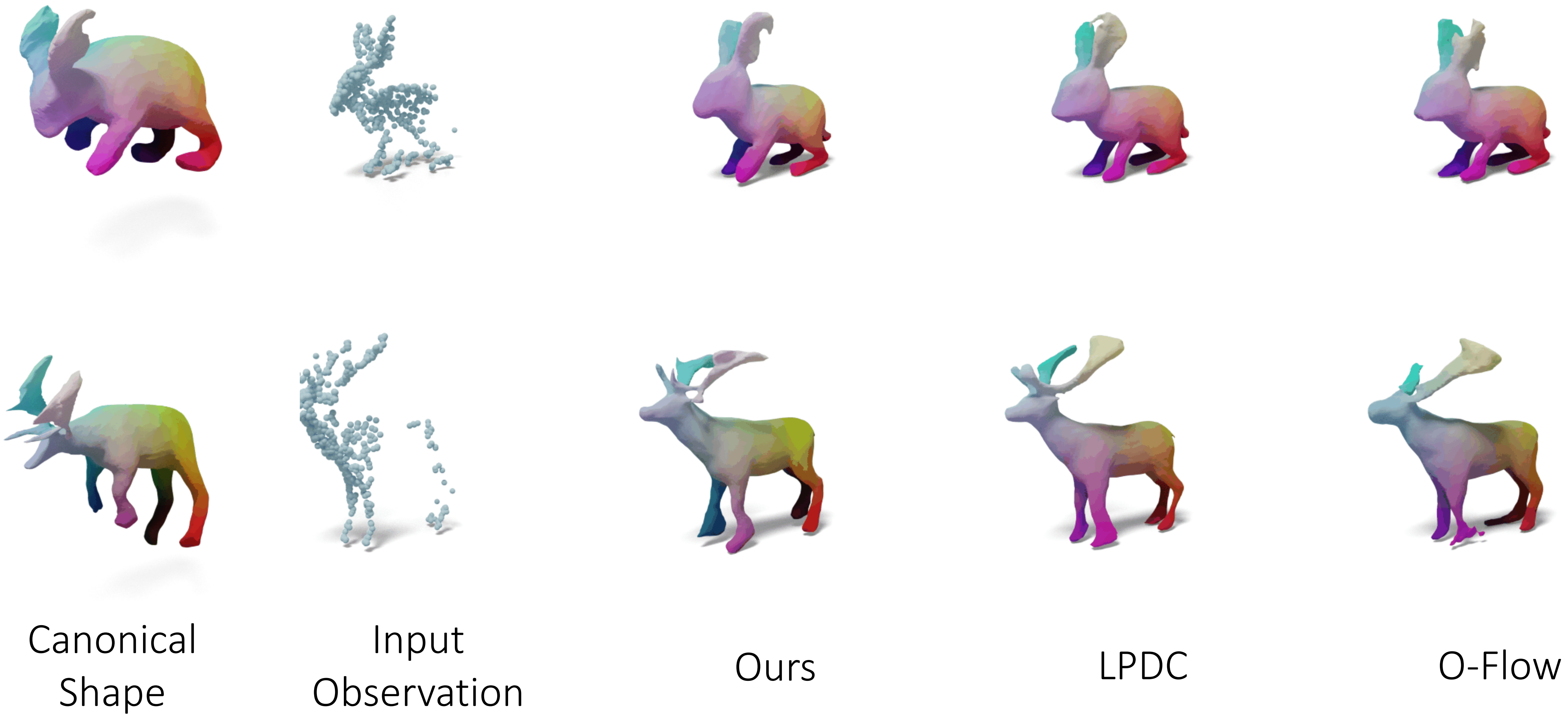


Comparison on DeformingThings4D Animals



Canonical Shape | Input Observation | Ours | LPDC | O-Flow

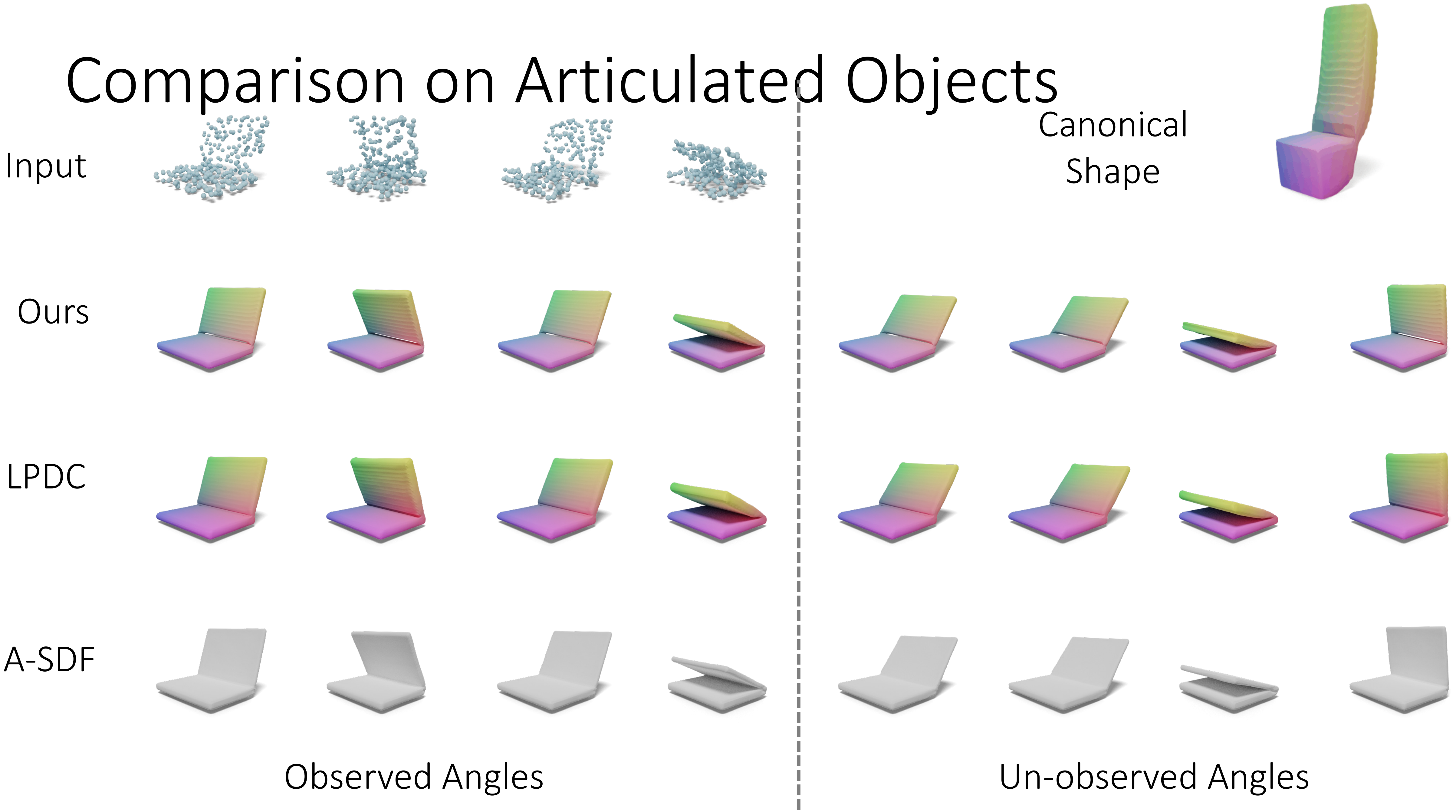
Comparison on DeformingThings4D Animals



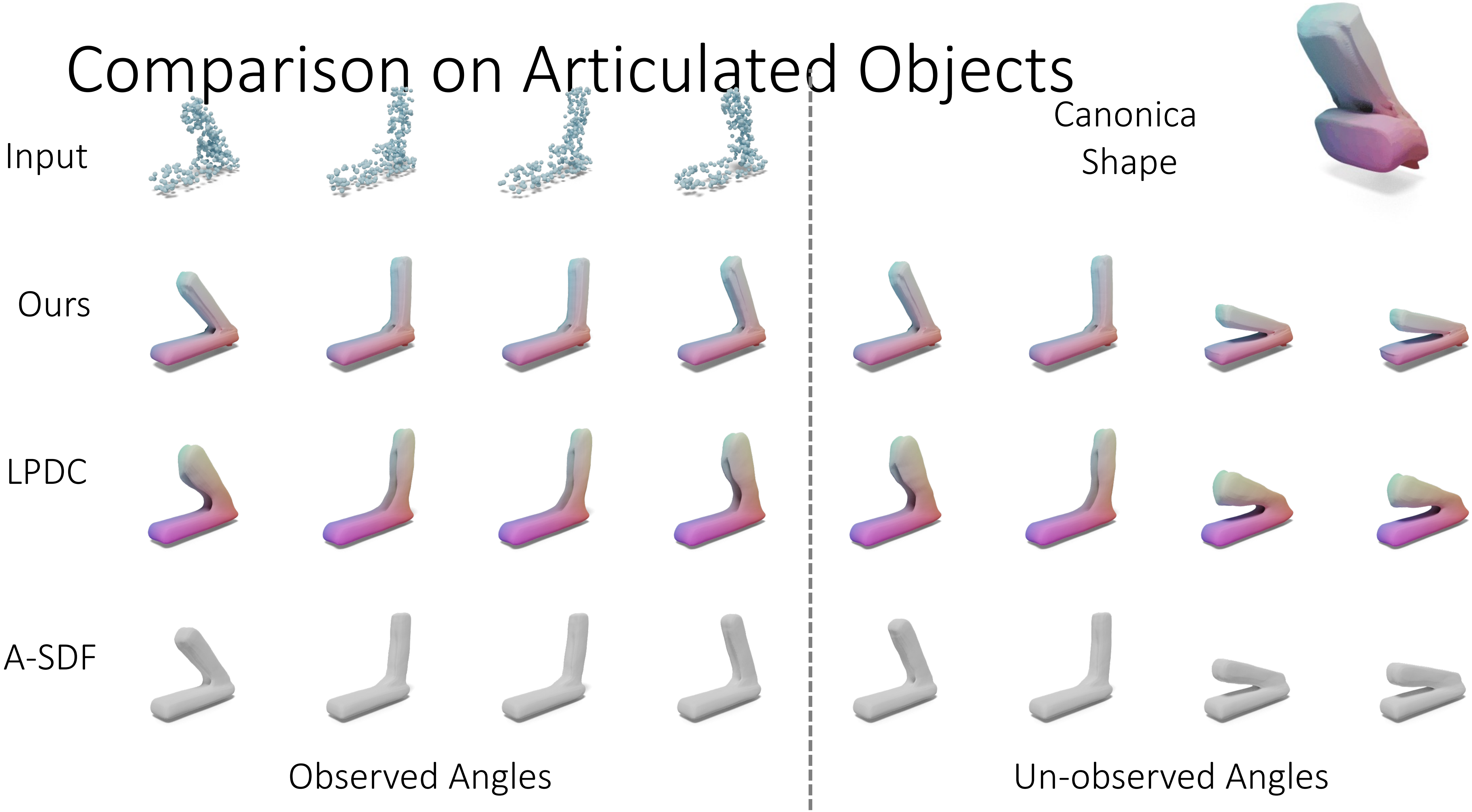
Comparison on DeformingThings4D Animals

Canonical Shape | Input Observation | Ours | LPDC | O-Flow

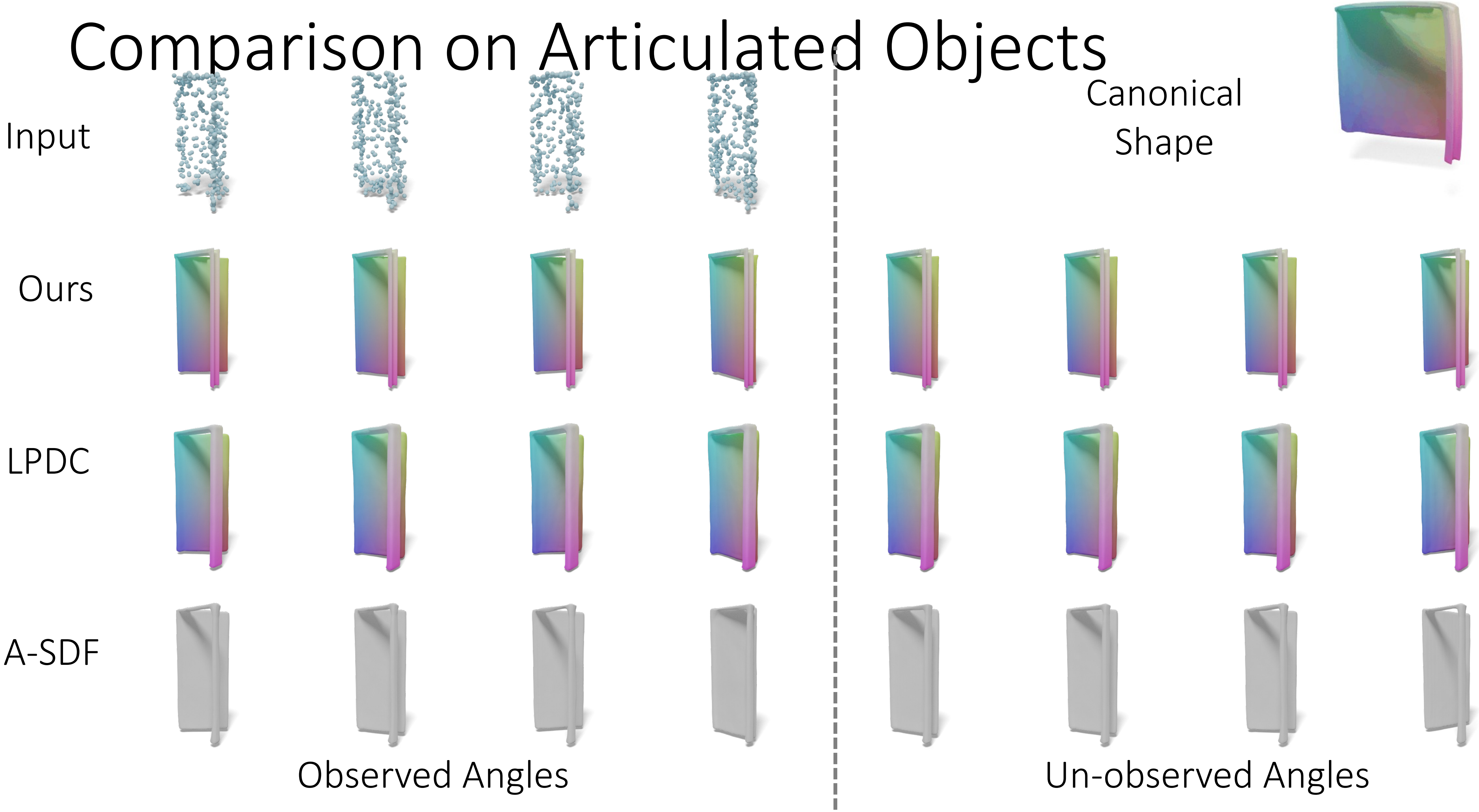
Comparison on Articulated Objects



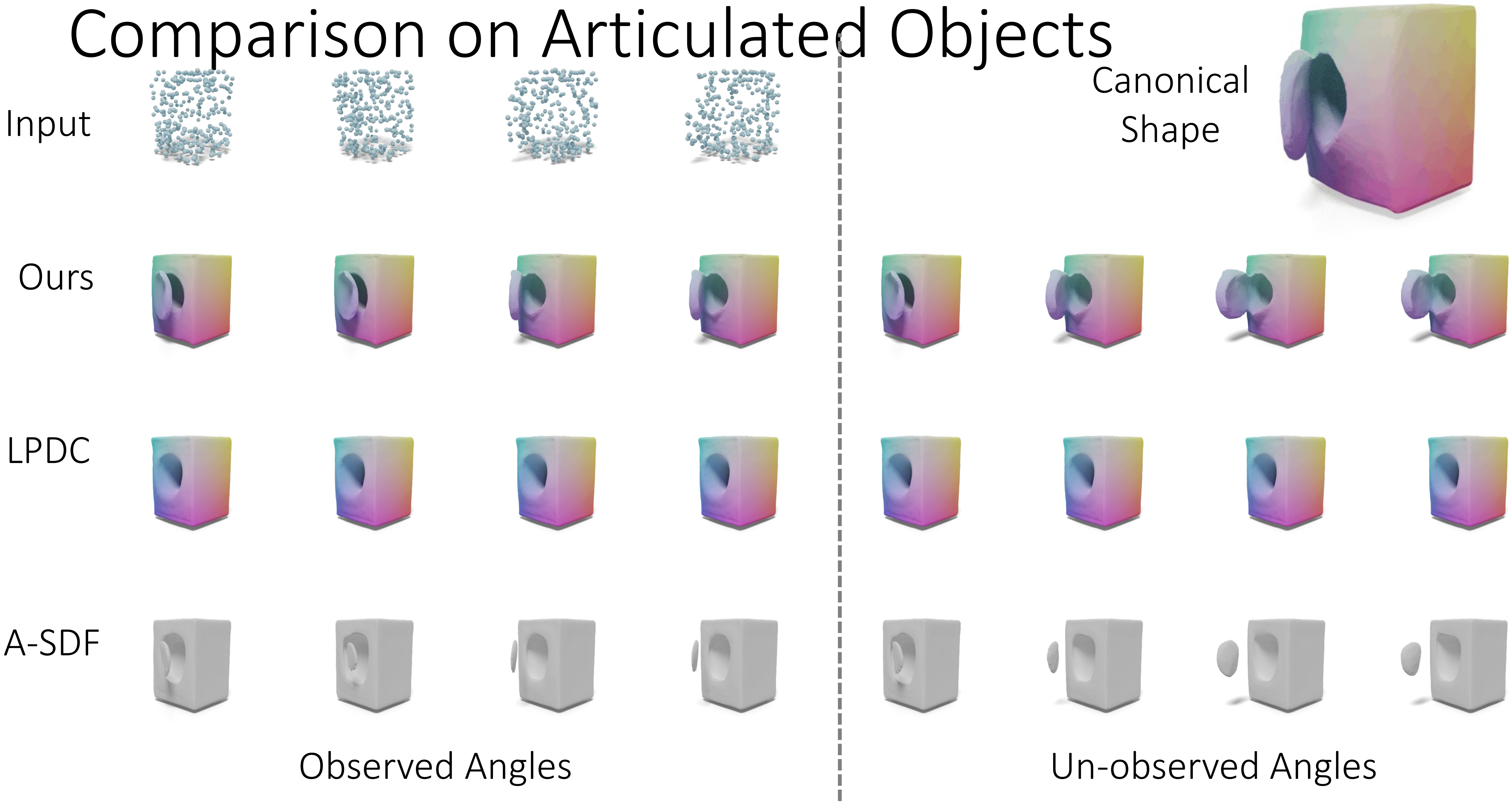
Comparison on Articulated Objects



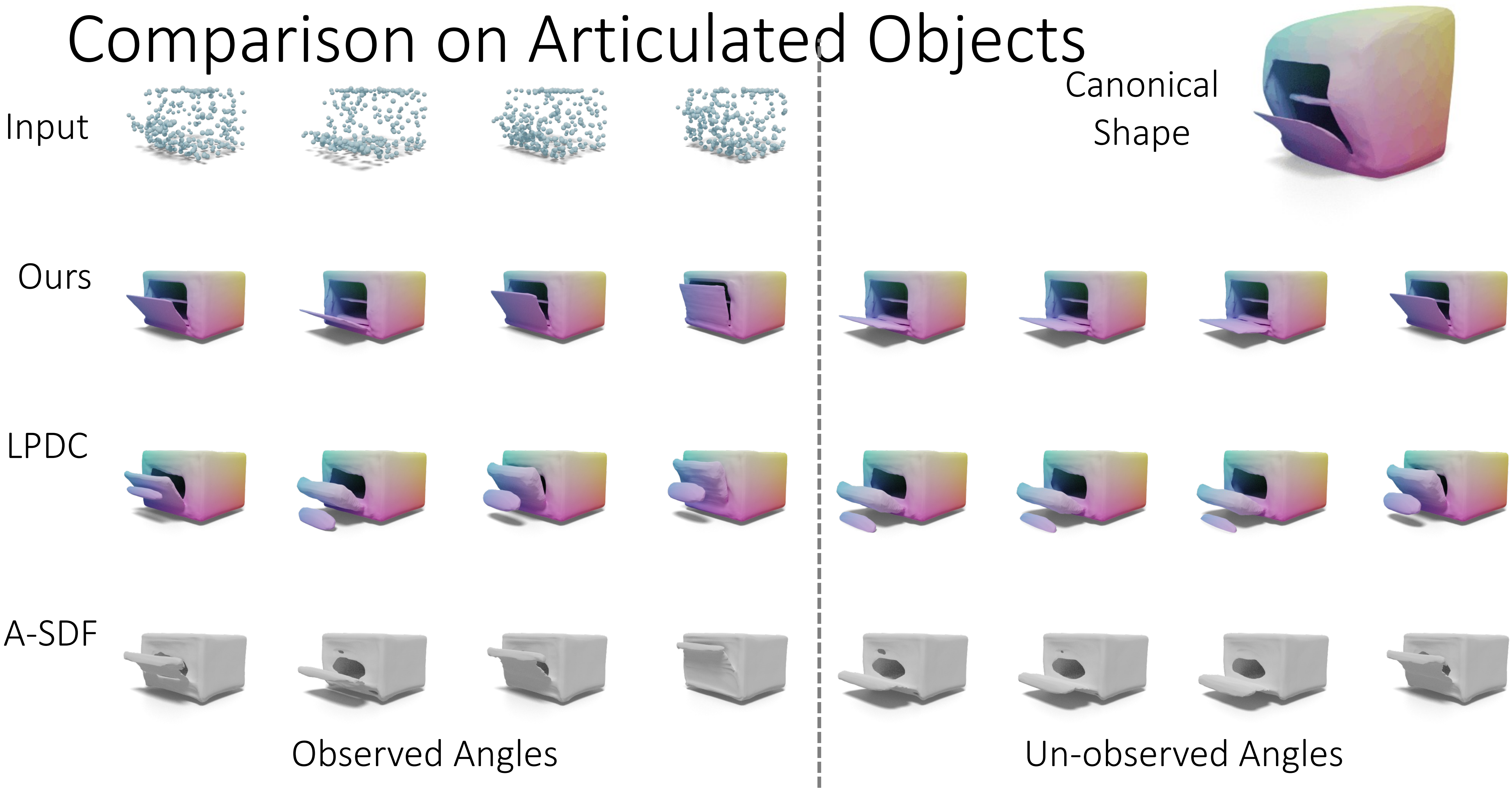
Comparison on Articulated Objects



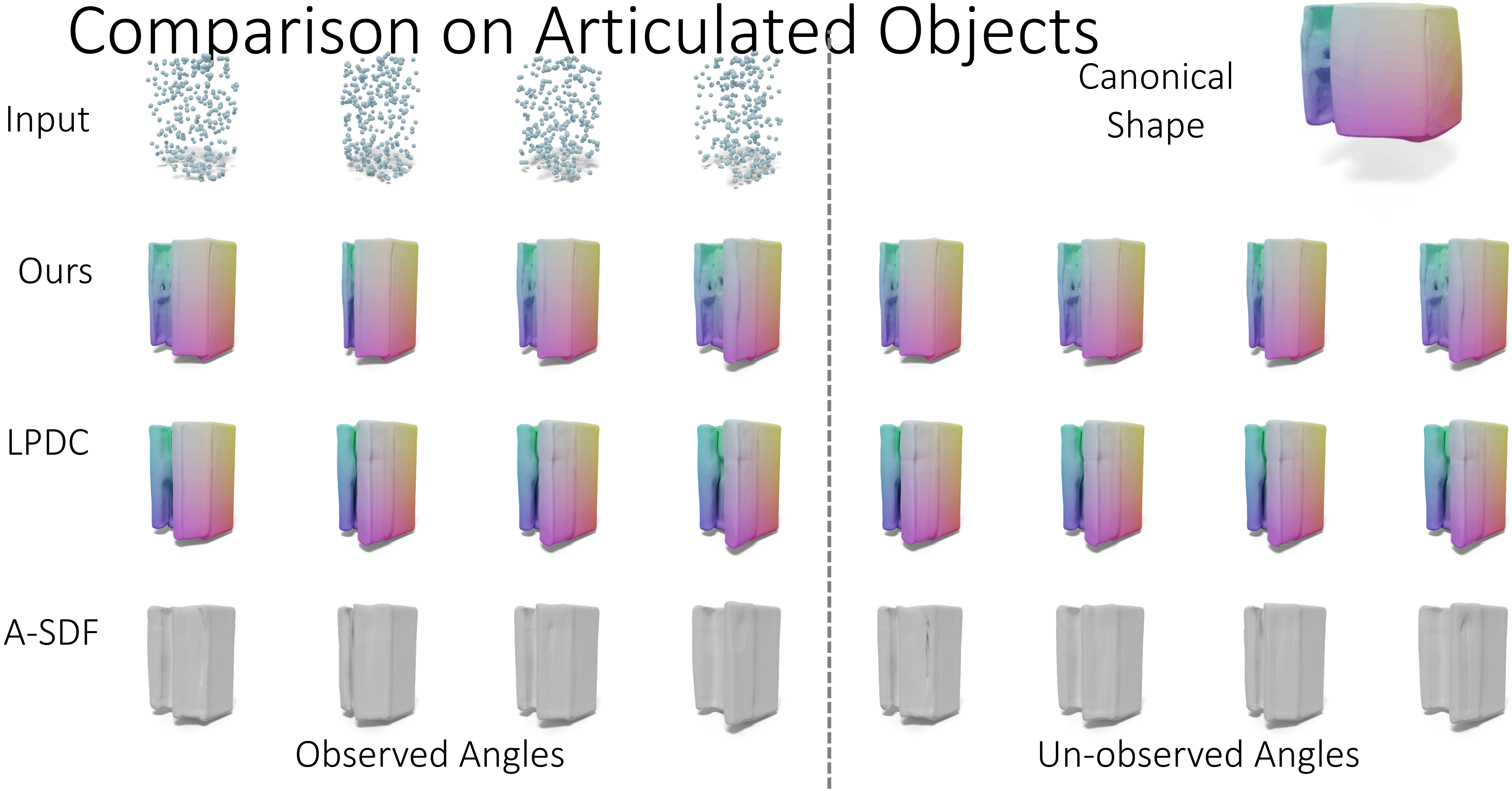
Comparison on Articulated Objects



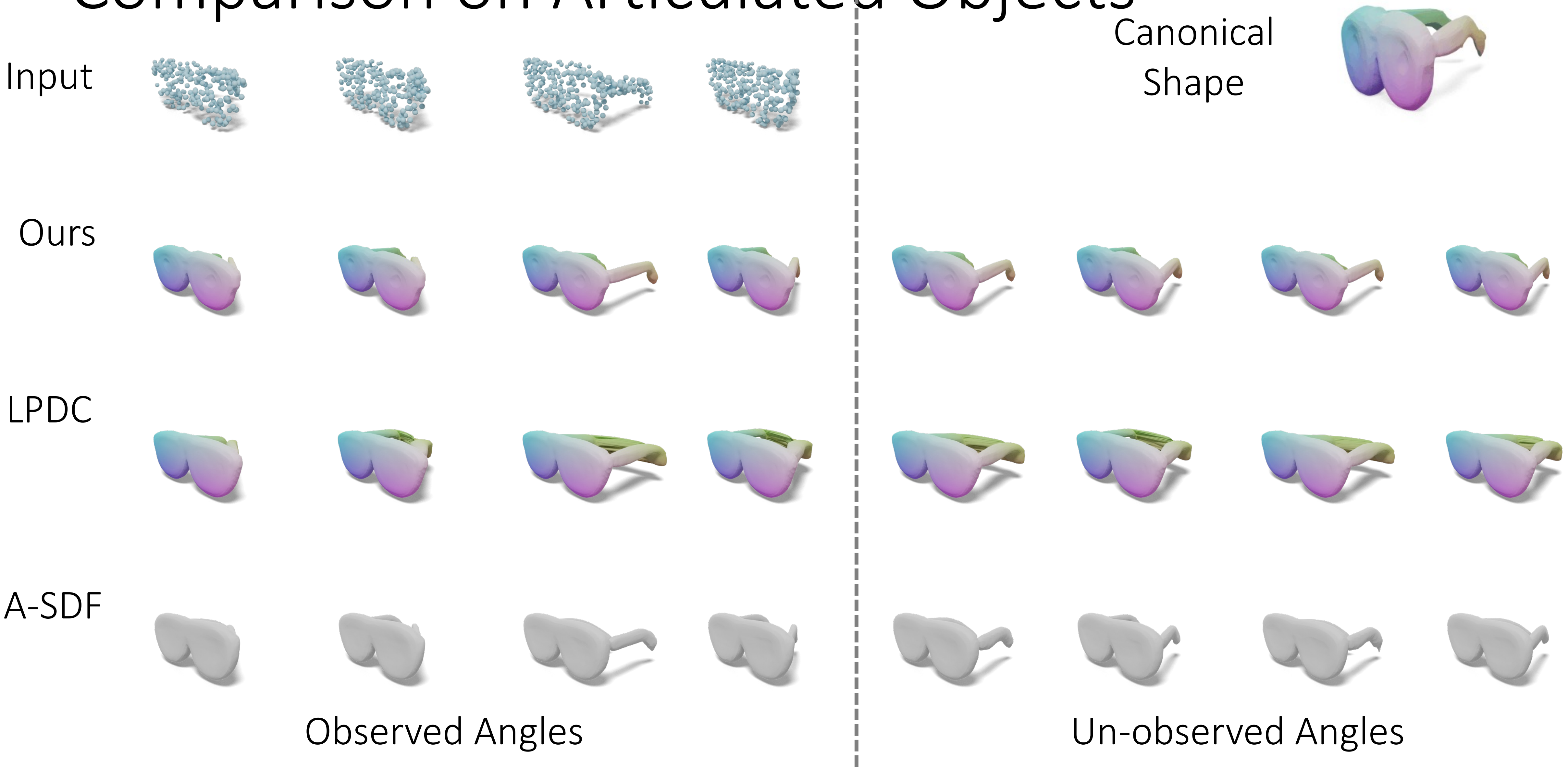
Comparison on Articulated Objects



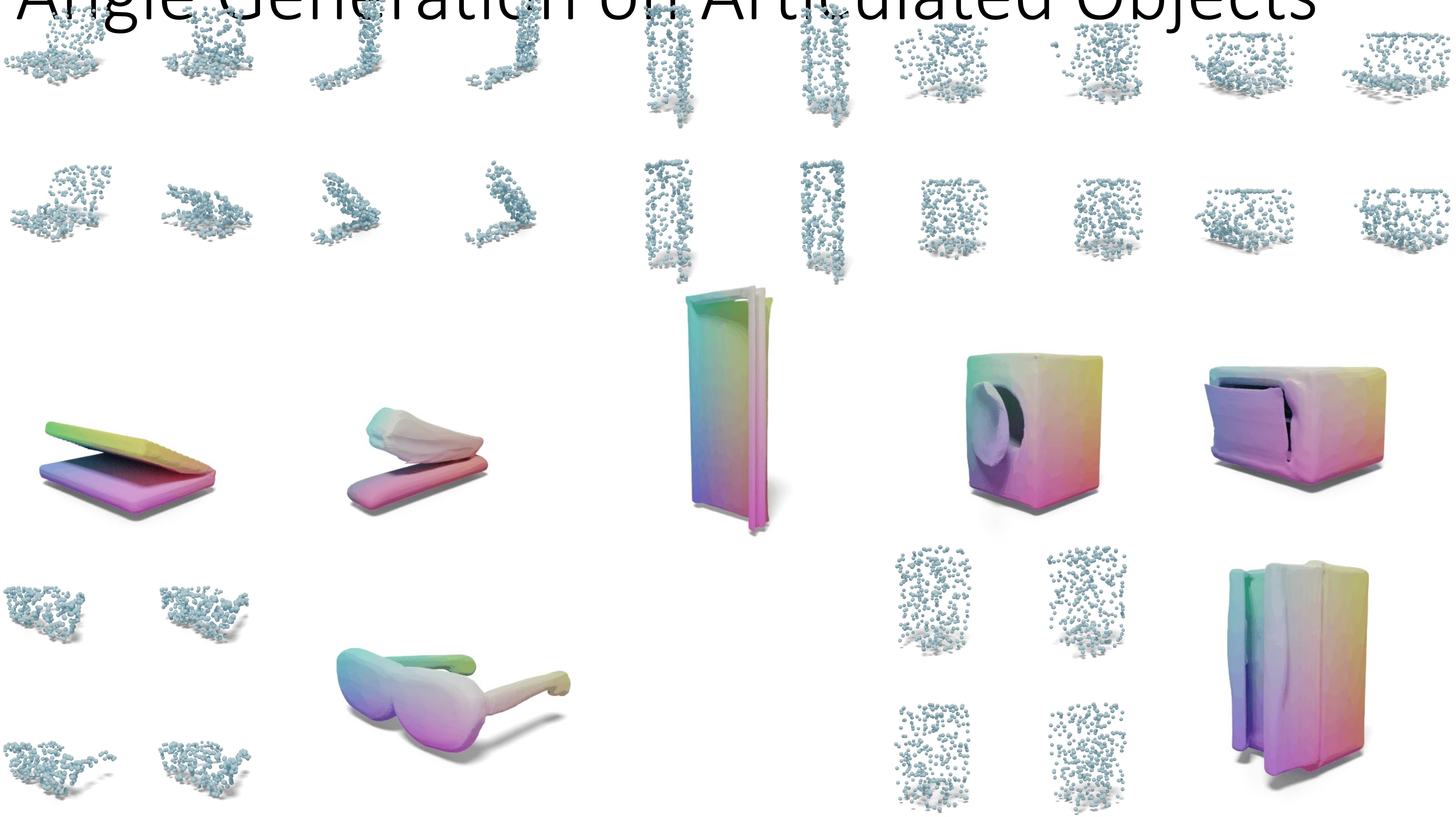
Comparison on Articulated Objects



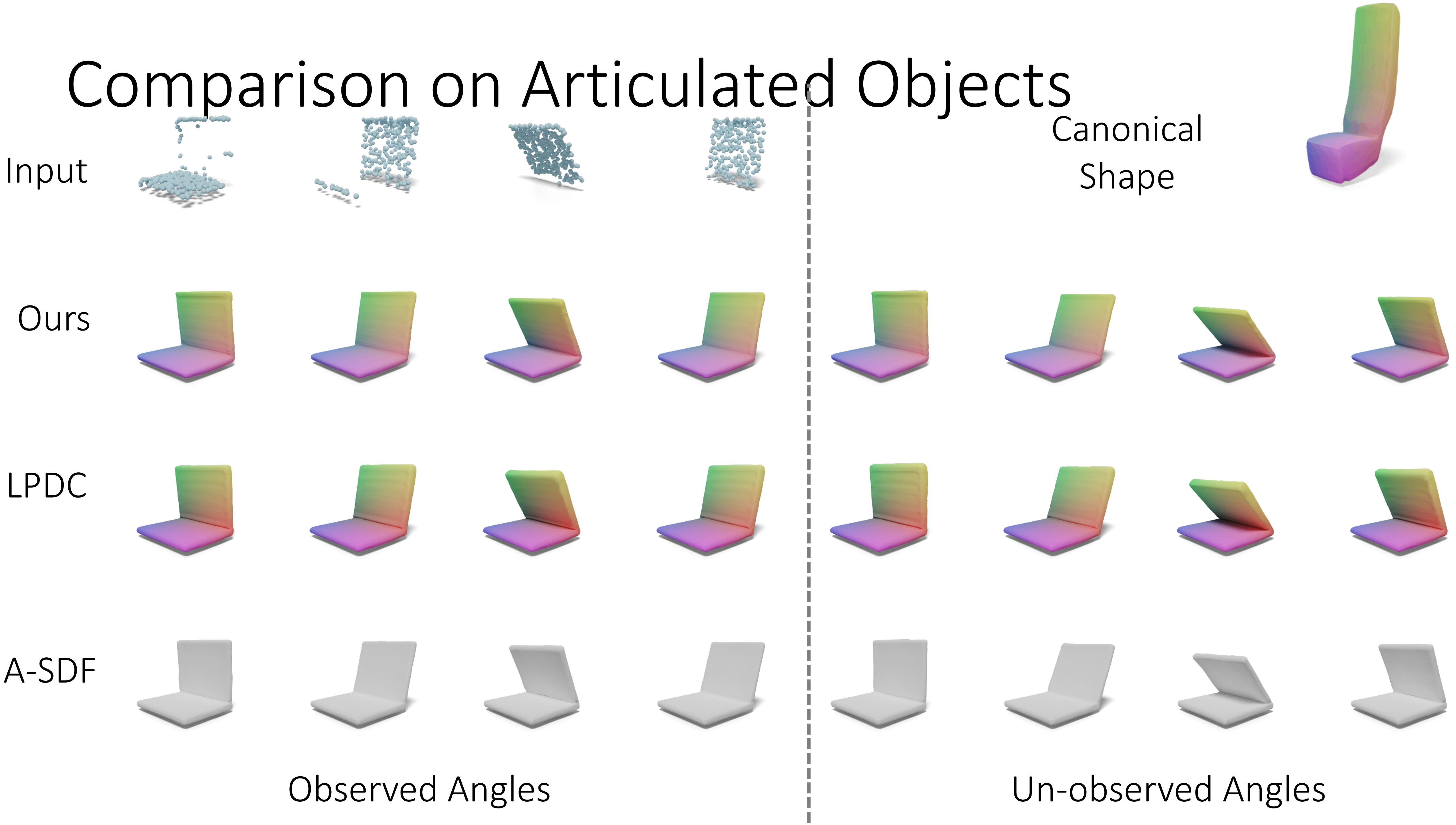
Comparison on Articulated Objects

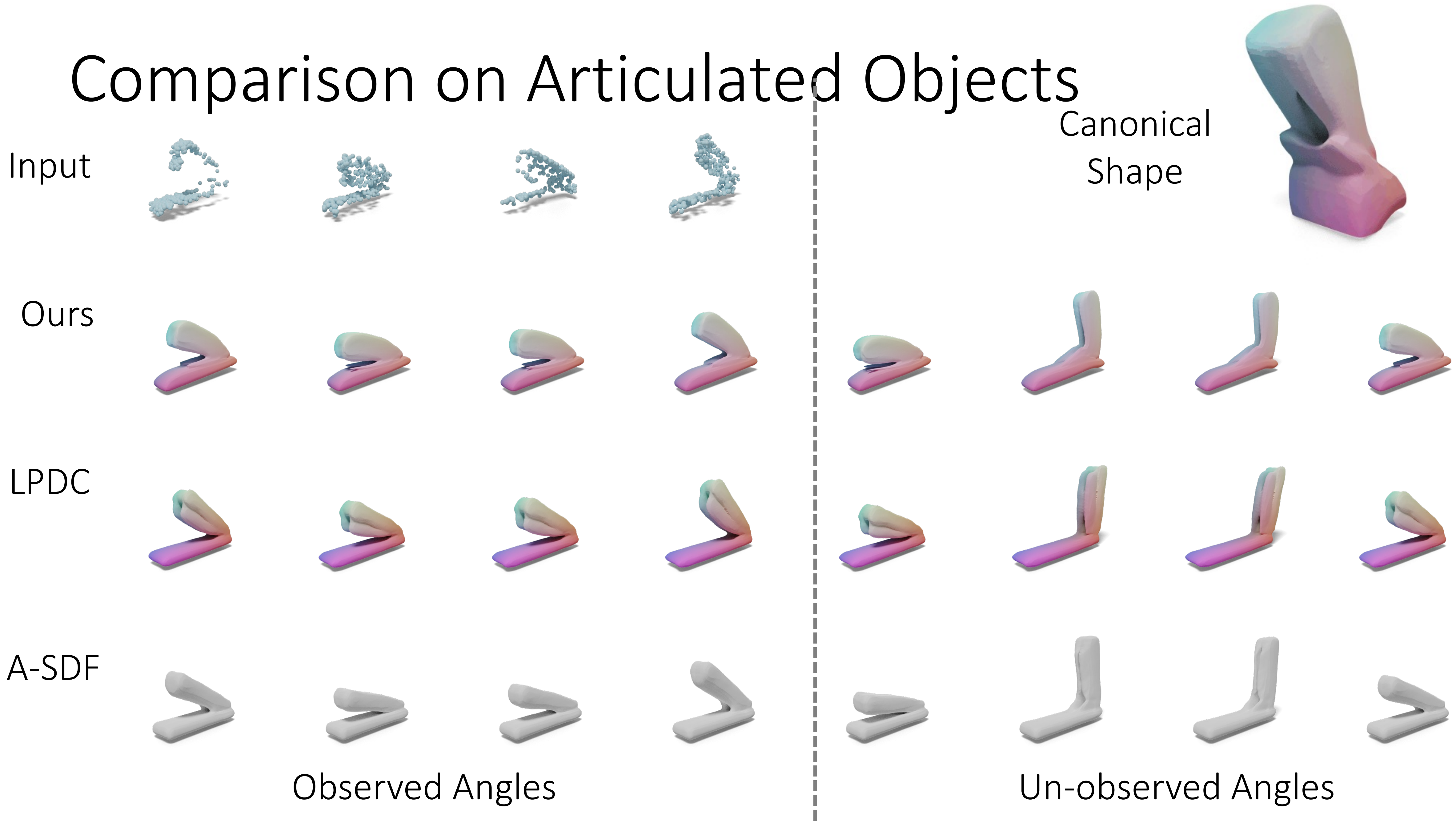


Angle Generation on Articulated Objects

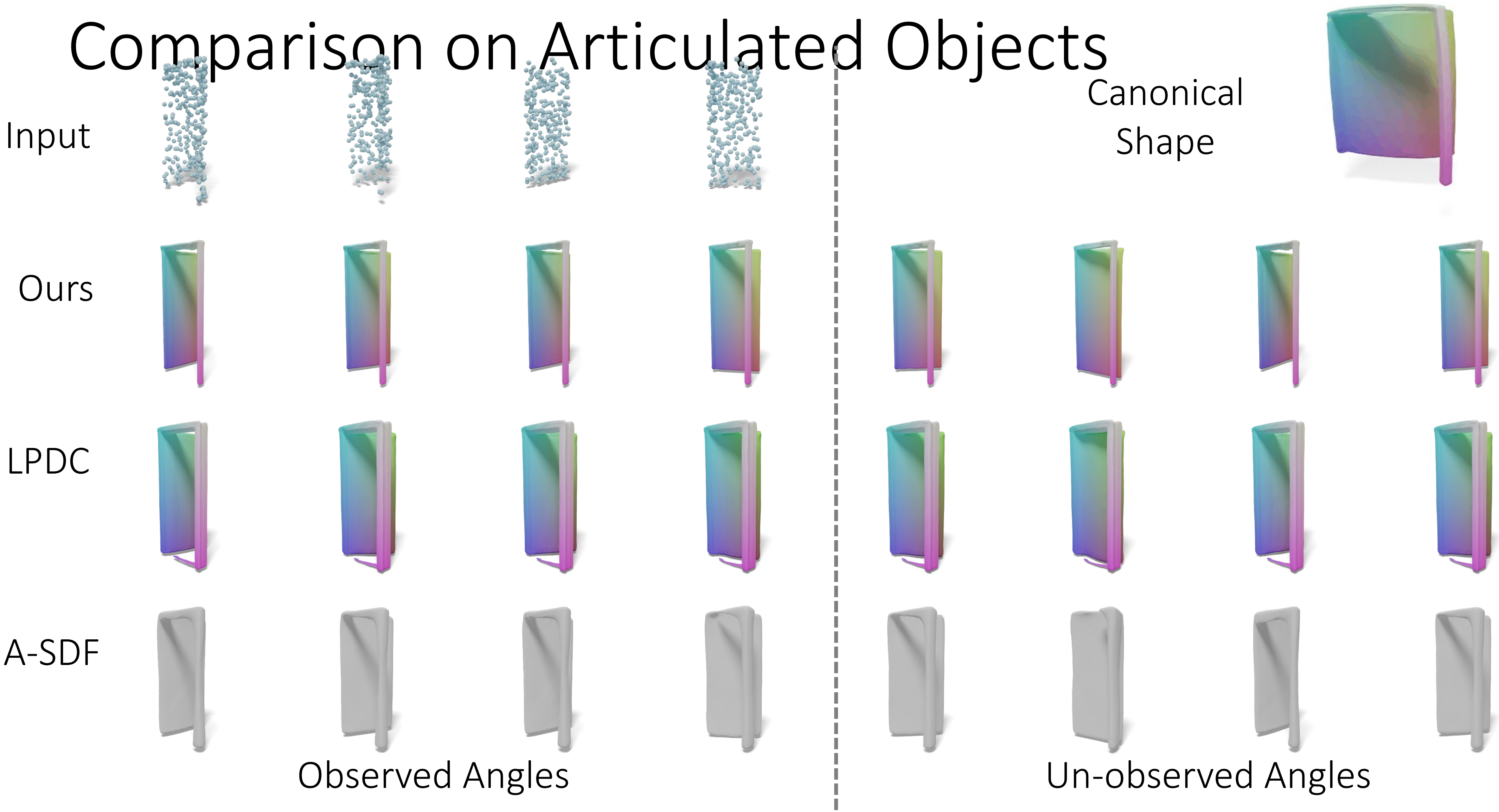


Comparison on Articulated Objects

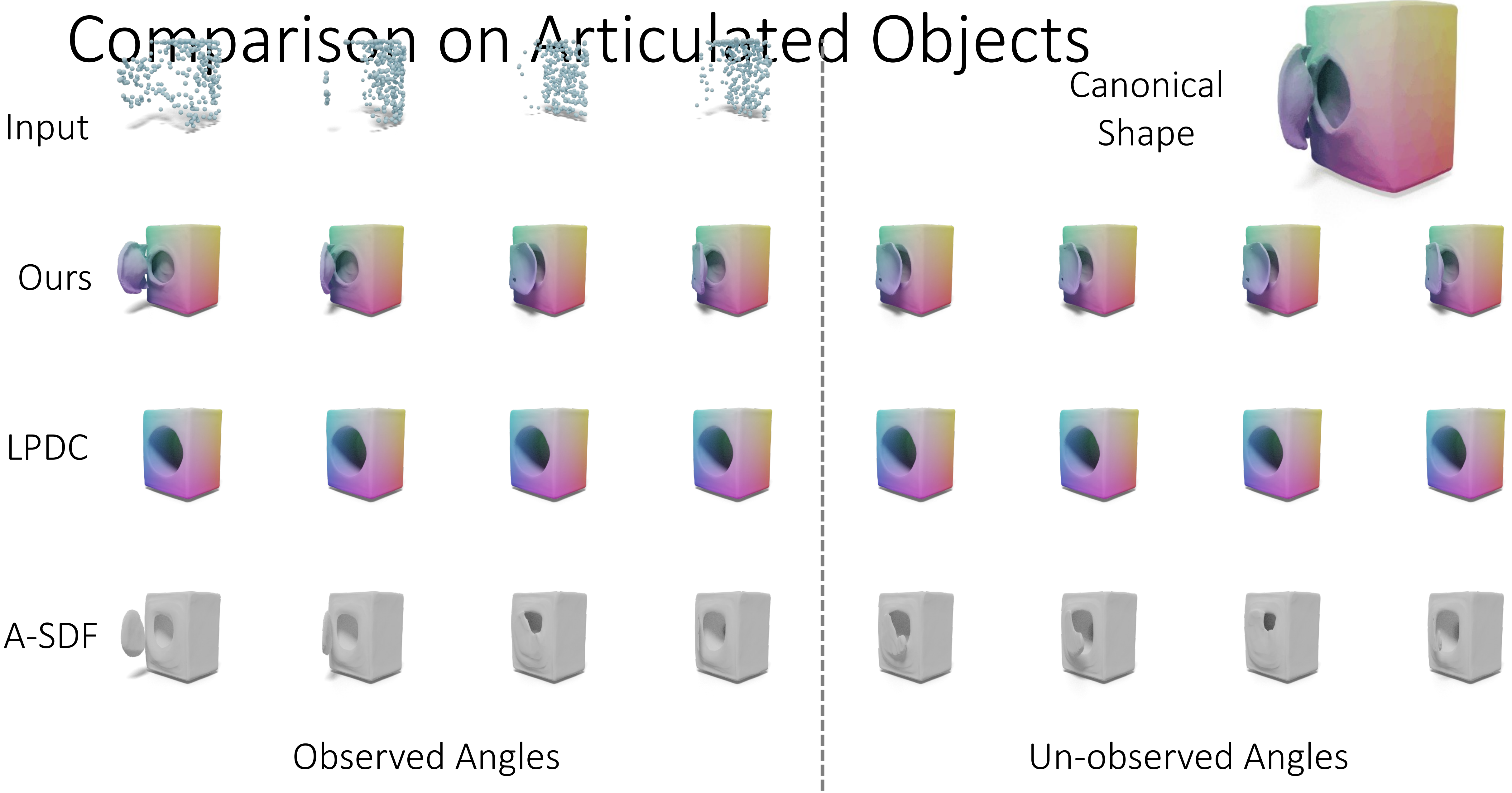



Comparison on Articulated Objects
Canonical Shape
Input
Ours
LPDC
A-SDF
Observed Angles
Un-observed Angles

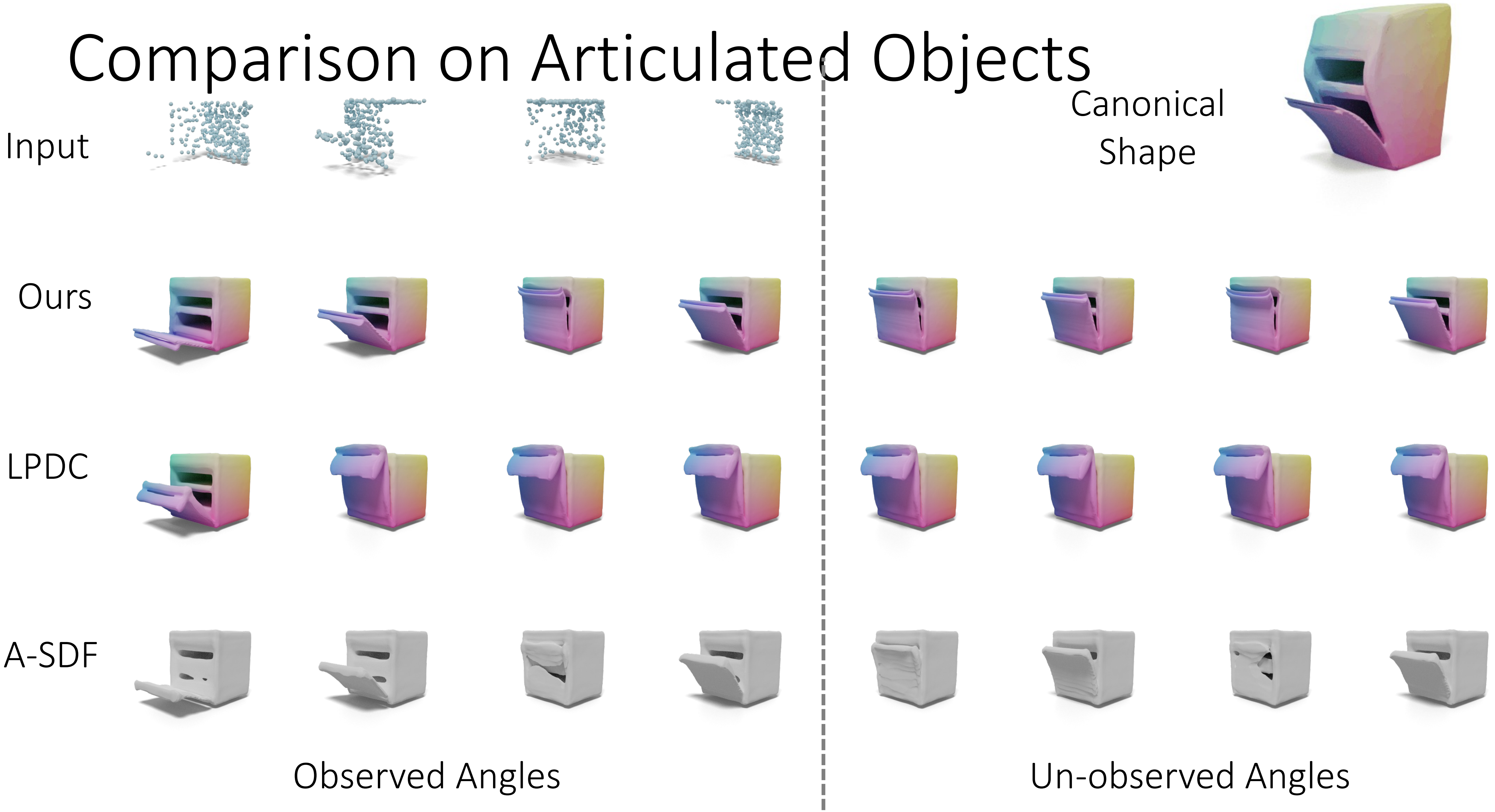
Comparison on Articulated Objects



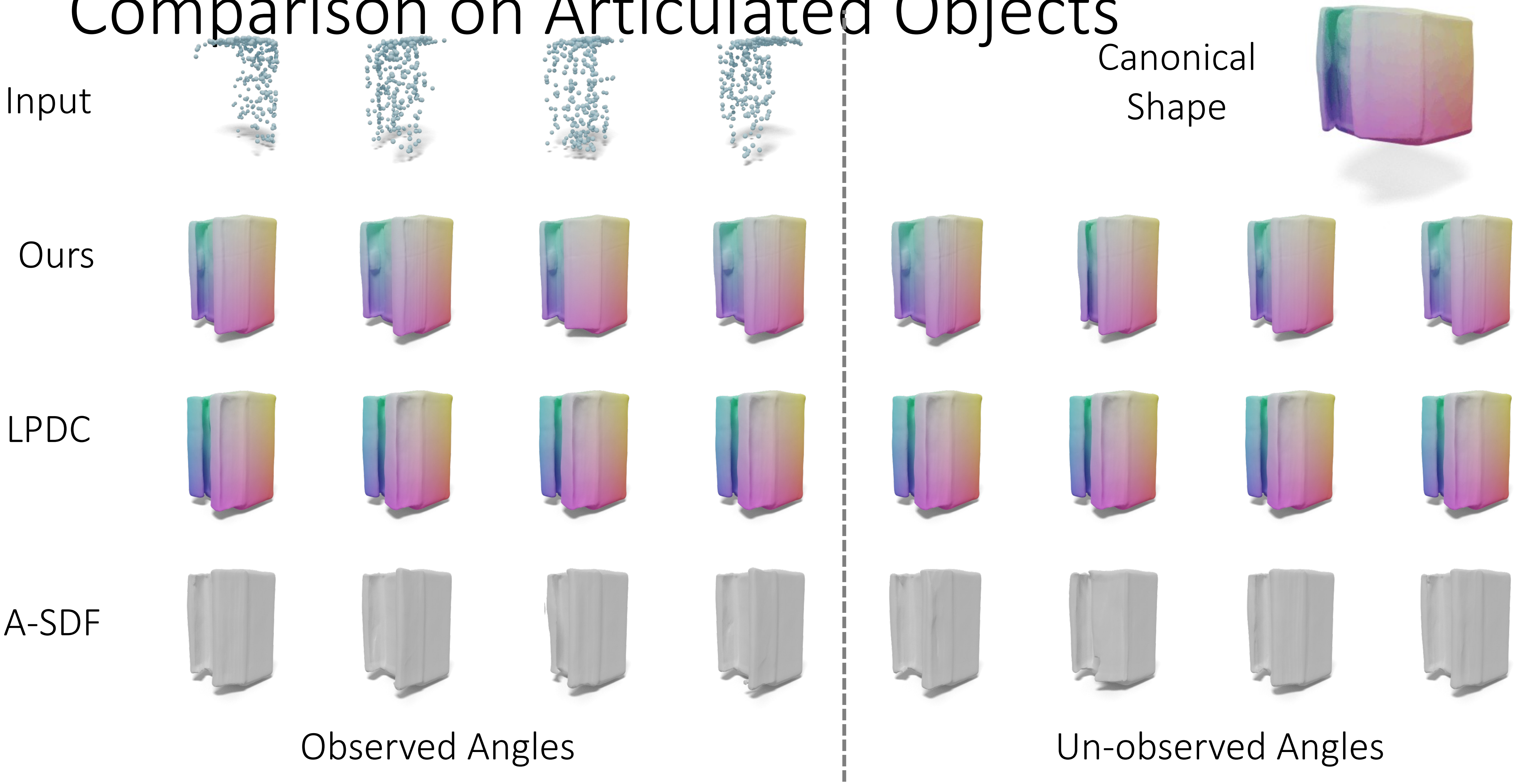
Comparison on Articulated Objects



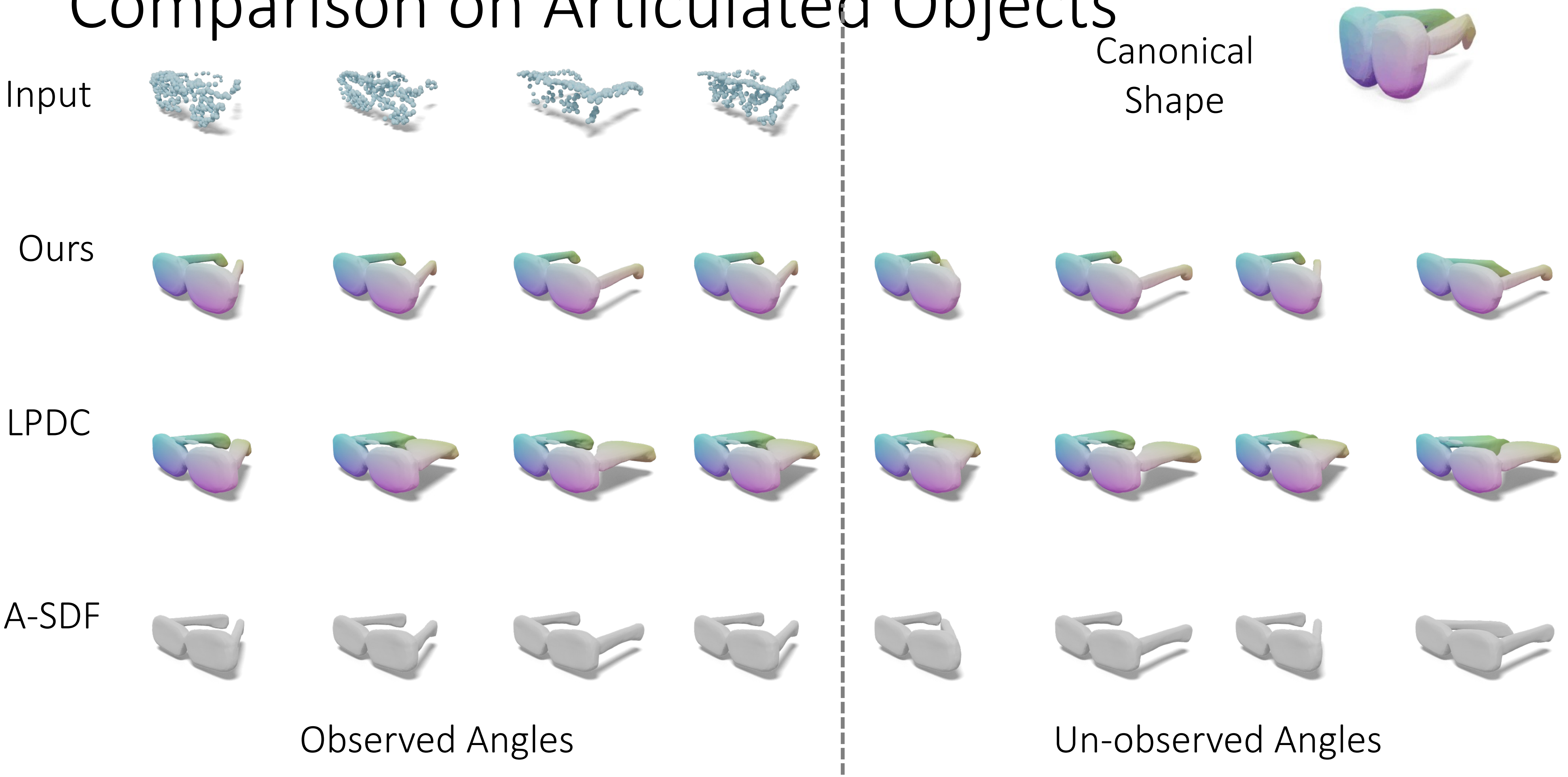
Comparison on Articulated Objects



Comparison on Articulated Objects

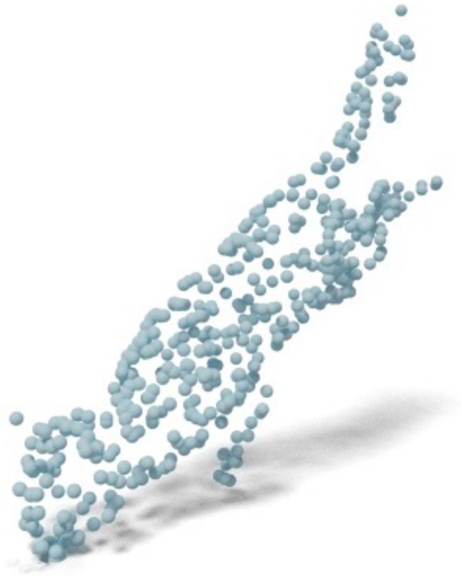


Comparison on Articulated Objects



Limitations

Since we use one single model weight for all animal categories, rare motions or instances sometimes can not be handled well.

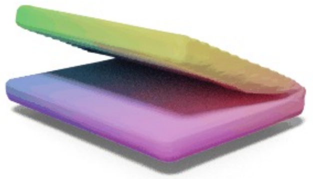


Topology Changes

CaDeX: Learning Canonical Deformation Coordinate Space for Dynamic Surface Representation via Neural Homeomorphism

Thanks for watching!

More details are in our paper!



CaDeX++: Fast and Robust AnyPoint Tracking

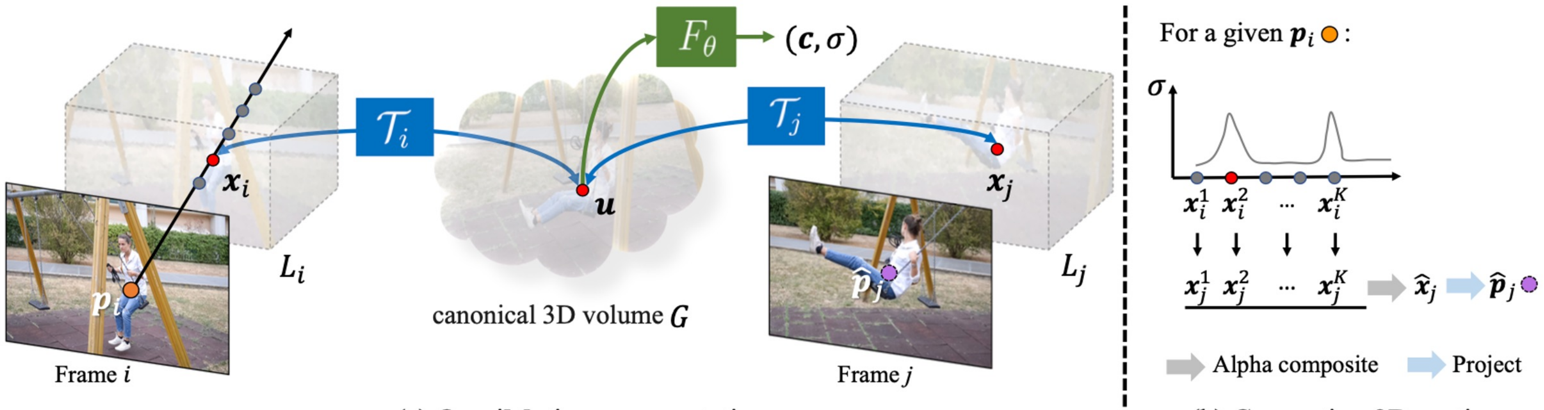
In Progress

Long Term Tracking

Baselines

Optim-based: Omnimotion

Learning geometry from pure 2D input



(a) OmniMotion representation

(b) Computing 2D motion

FeedForward: Cotracker …

Traditional 2D feature-based tracking

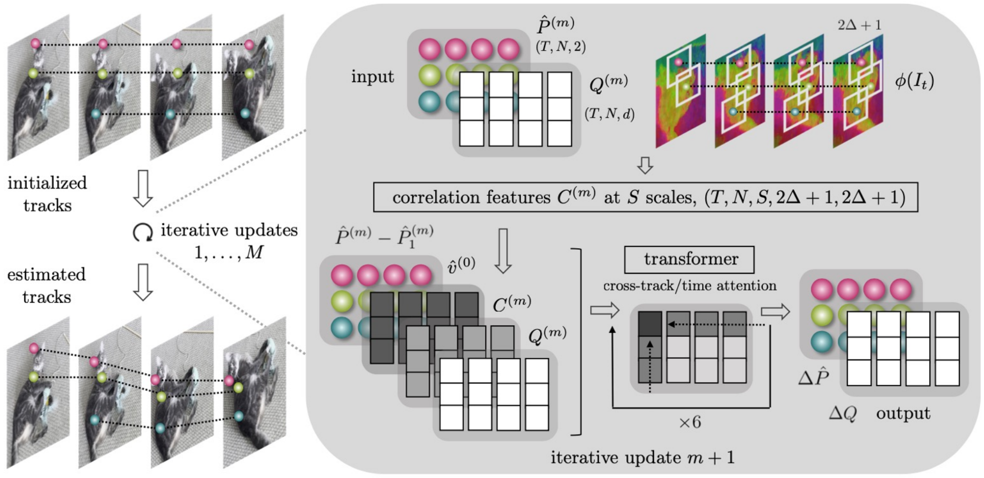
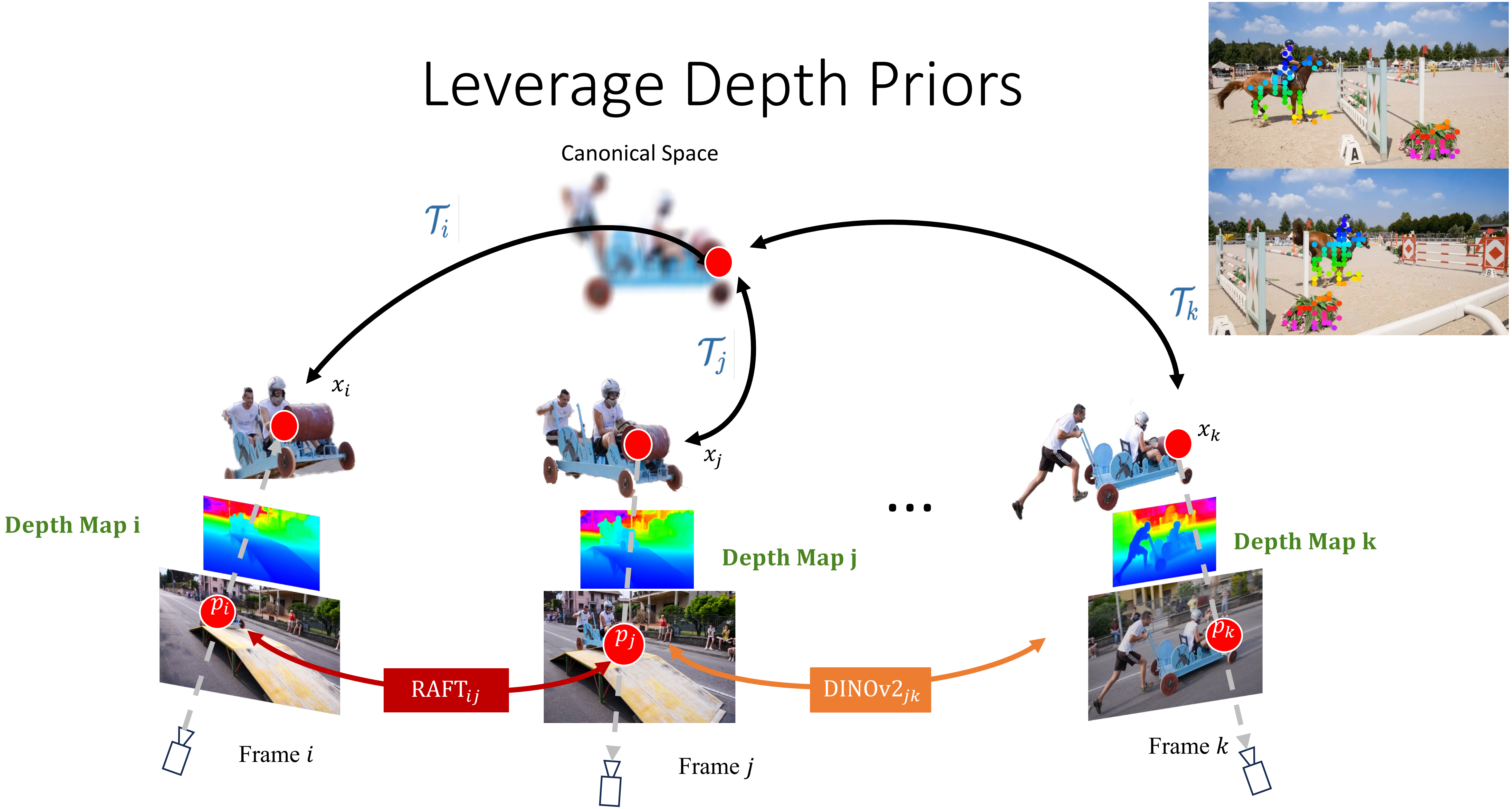


Figure 3. **CoTracker architecture.** Visualization of one sliding window with M iterative updates. During one iteration, we update point tracks $\hat{P}^{(m)}$ and track features $Q^{(m)}$. $Q^{(0)}$ is initialized with the initially sampled features Q for all sliding windows, $\hat{P}^{(0)}$ with the starting locations for the first window. For other windows, $\hat{P}^{(0)}$ starts with predictions for frames processed in the preceding sliding window, and with the last predicted positions for the unseen frames. We compute visibility $\hat{v}$ after the last update M.

Motivations

- Omnimotion has several drawbacks:
  - Extremely Slow
  - The 3D geometry is weak
  - Only take short term optical flow as local information
  - Weak robustness
- Our contribution:
  - Better Deformation Homeomorphism: Locality and Non-Linearity.
  - Explicit take mono-depth into the model, introduce more 3D inductive bias.
  - Exploit the DINO information in long term.

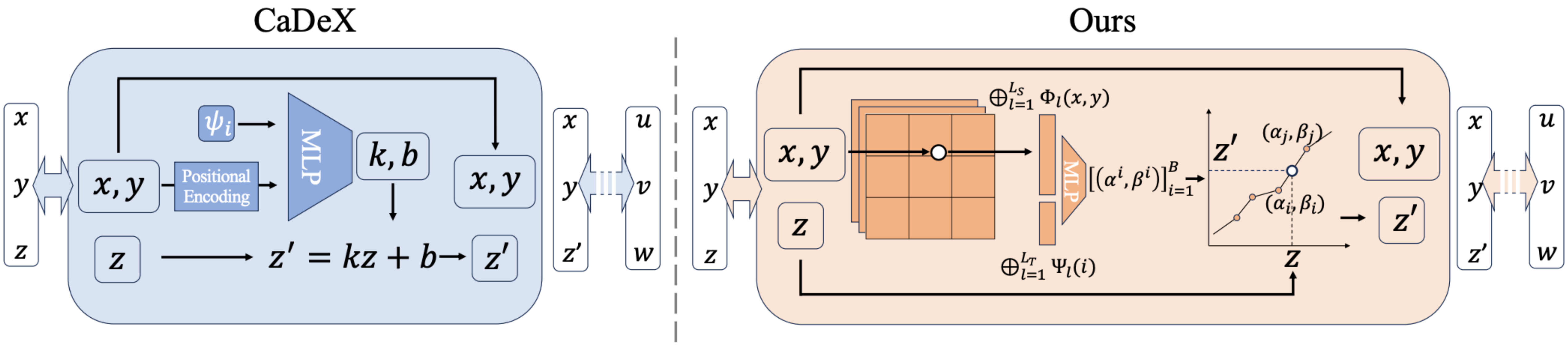
Leverage Depth Priors



Ours: Locality of the Deformation Network

Baseline: a large MLP + linear affine deform each layer

Ours: Local-feature & small MLP + nonlinear deform



Ours: Long Term Supervise from foundational features

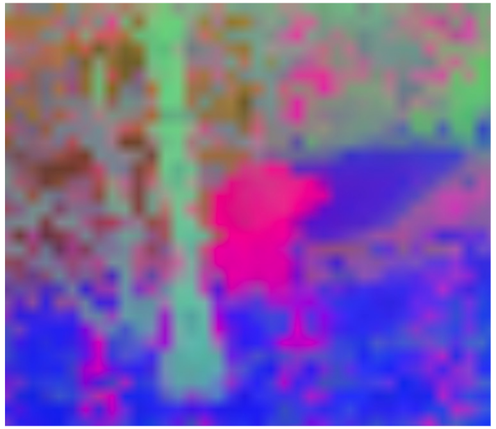
Optical flow: **dense**, but **“cut” by occlusion**

Long term info: global **coarse** feature matching

PCA vis of features

Flow: Fail

Feature Match: Success



Ours: Long Term Supervise from foundational features

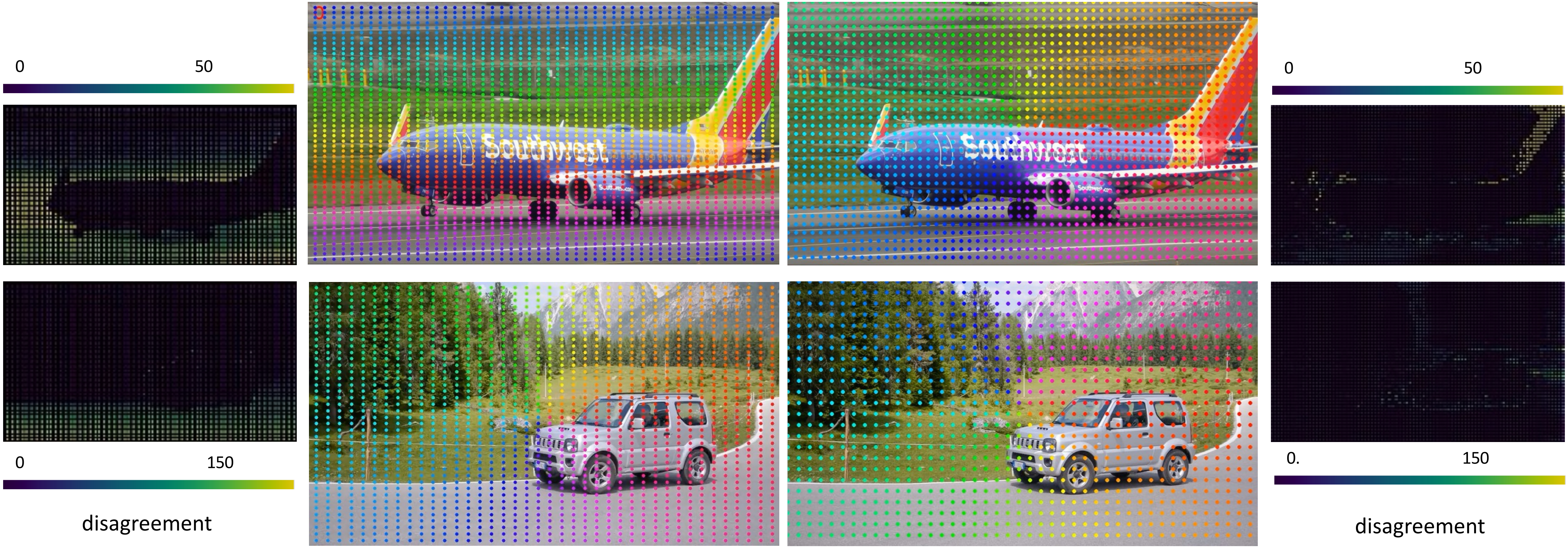
CoTracker: Fail in texture-less area

CoTracker: trajectory disagree with optical flow on texture-less points

CoTracker Ours

CoTracker: Fail in texture-less area

CoTracker: trajectory disagree with optical flow on texture-less points

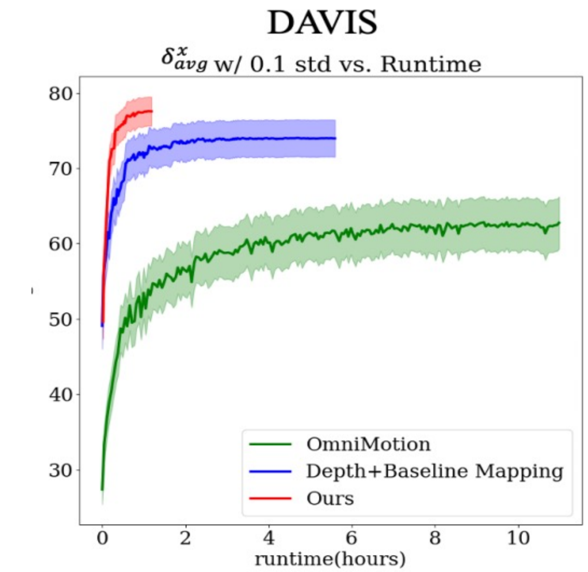


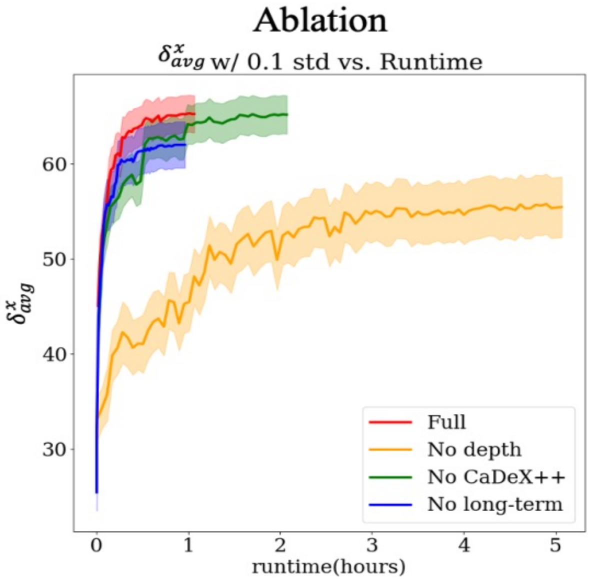
CoTracker Ours

Result

Result: Performance

	Method	DAVIS				RGB-Stacking			
		AJ↑	δ^x_{avg} ↑	OA↑	TC ↓	AJ ↑	δ^x_{avg} ↑	OA↑	TC ↓
Feed-forward	PIPs [11]	39.9	56.0	81.3	1.78	37.3	50.6	89.7	0.84
	Flow-Walk [3]	35.2	51.4	80.6	**0.90**	41.3	55.7	92.2	**0.13**
	MFT [24]	56.1	70.8	86.9	-	-	-	-	-
	TAP-Net [8]	38.4	53.4	81.4	10.82	61.3	73.7	91.5	1.52
	TAPIR [9]	59.8	72.3	87.6	-	**66.2**	77.4	**93.3**	-
	CoTracker [14]	**65.1**	**79.0**	**89.4**	0.93	65.9	**80.4**	85.4	0.14
Opti-mization	Connect RAFT [33]	30.7	46.6	80.2	0.93	42.0	56.4	91.5	0.18
	Deformable Sprites [40]	20.6	32.9	69.7	2.07	45.0	58.3	84.0	0.99
	OmniMotion [34]	51.7	67.5	85.3	0.74	**77.5**	87.0	93.5	**0.13**
	Ours	**59.4**	**77.4**	**85.9**	**0.68**	75.4	**87.1**	**93.6**	0.15



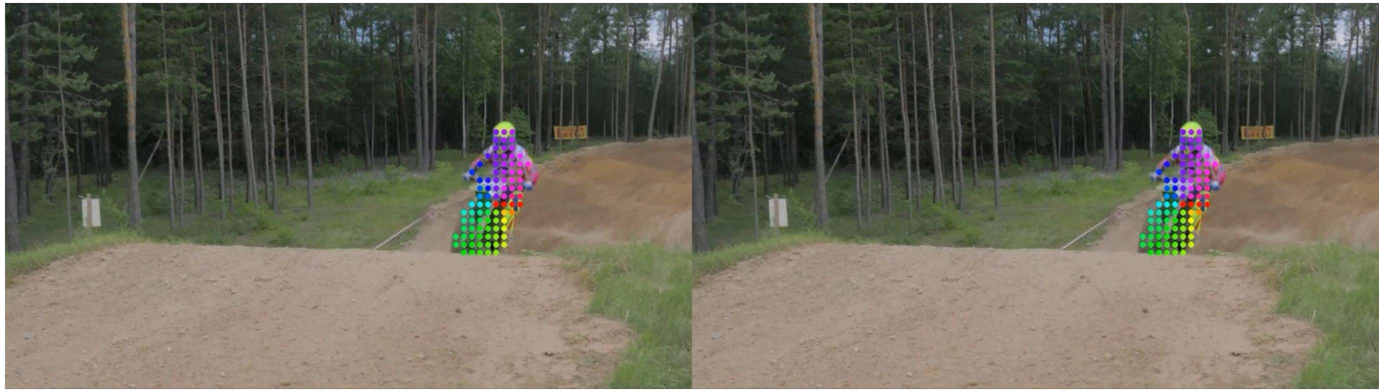


Result: Robustness

Comparison of convergence robustness

Method	$\delta^x_{avg} \uparrow$							
	motocross-jump				libby			
	min	max	mean	std	min	max	mean	std
Omnimotion	4.7	60.5	26.3	26.1	2.3	18.0	8.86	5.9
Ours w/o depth	4.4	65.5	44.3	23.5	1.8	20.2	12.7	6.6
Ours	75.2	76.4	75.6	0.5	40.1	48.5	45.7	3.0

Example of convergence and divergence



Ours: robust

Omnimotion: prone to fail

Summary

- Lift 2D video to 3D scene
- Locality & Non-linear deformation
- Long-term DinoV2 correspondence

- Low GPU memory consumption (Omnimotion: >10G, Ours: 3G on DAVIS)
- Fast
- Robust
- Performance gain (better than Omnimotion, comparable with feed-forward methods)

Limits

- Scene-Sensitive (Optimization-based)
- No semantic similarity constraint
- Fitting time increases with video length

Represent, Reconstruct and Generate the 4D Real World

Jiahui Lei

2024 Sep