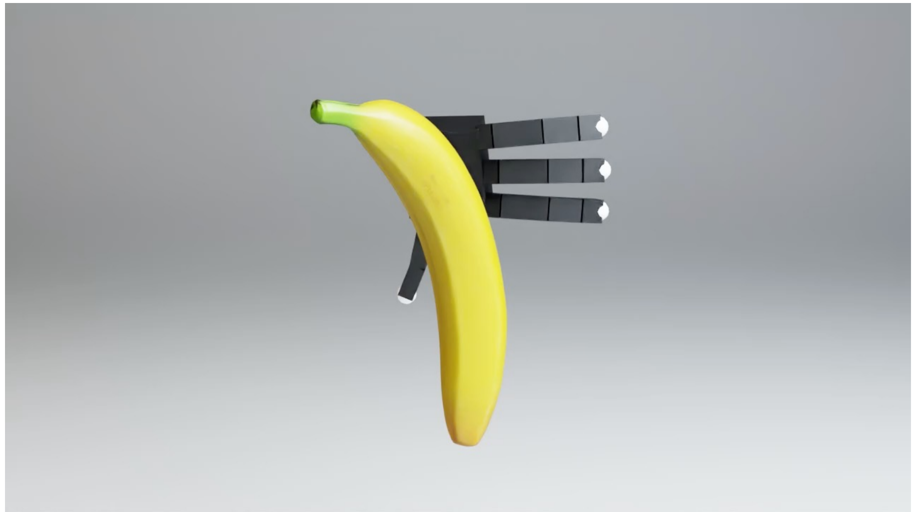
Robot Learning with Implicit Representations

Perception, Action, and Simulation



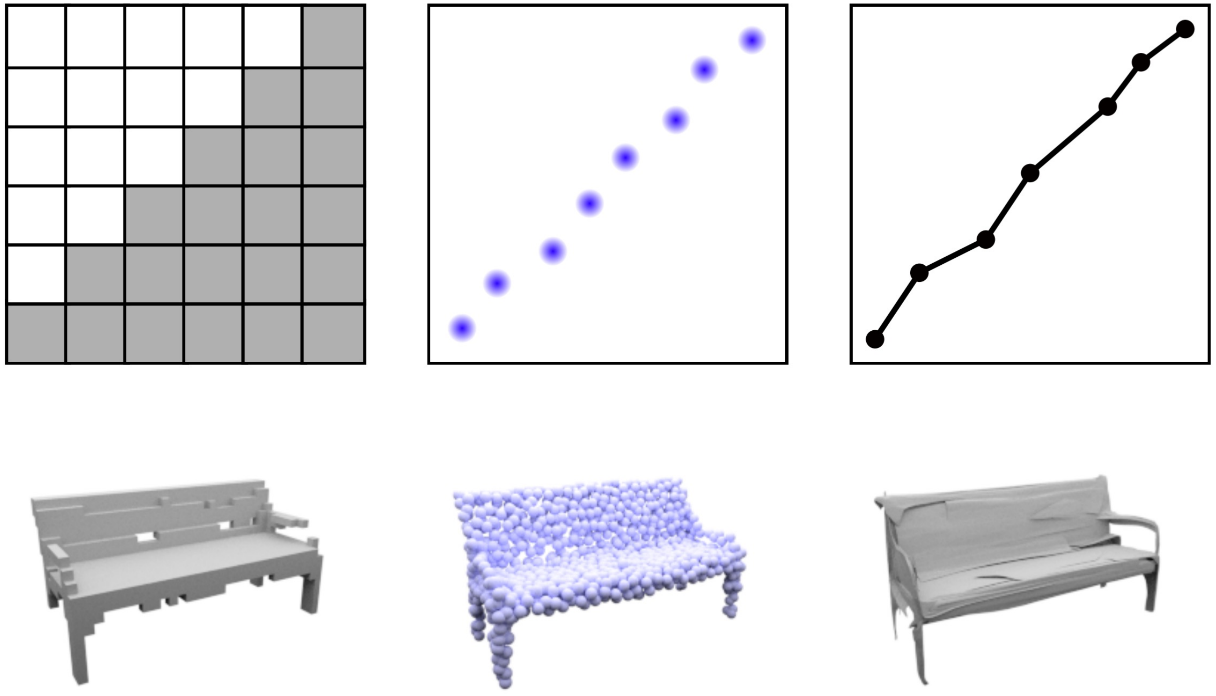
Animesh Garg

RSS 2022 Workshop

People AI & Robots pair
UNIVERSITY OF TORONTO
VECTOR INSTITUTE
NVIDIA

What is Implicit Neural Representation?

3D Representations in Visual Computing



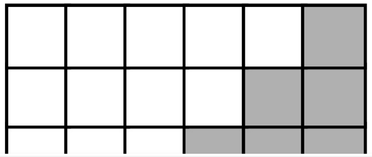
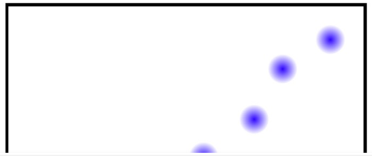
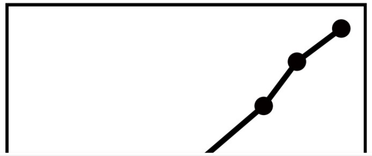
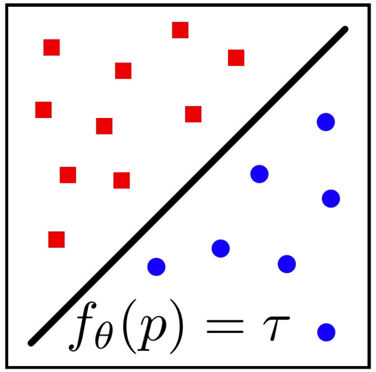
Voxels   Point Clouds   Mesh

✓ Discrete Representations
✓ Intuitive Spatial Map

✗ Memory
✗ Arbitrary Topologies
✗ Connectivity Structures

Occupancy Nets, CVPR 20

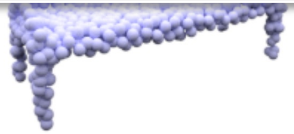
What is Implicit Neural Representation?

- ✓ Continuous Representations
- ✓ “Infinite” Spatial Resolution
- ✓ Memory depends on signal complexity

- ✗ Not Analytically Tractable

Voxels



Point Clouds

Mesh

Implicit Representation

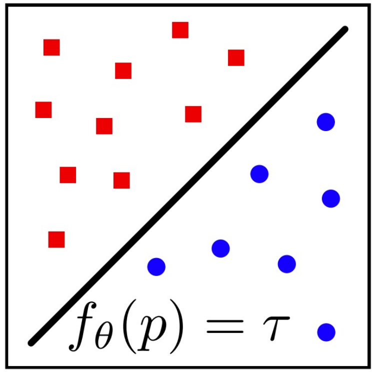
Occupancy Nets, CVPR 20

What is Implicit Neural Representation?

Coordinates — Values

$$f : \mathbb{R}^D \rightarrow \mathbb{R}^C$$

$$V = f(r)$$



Images:

r: (x, y), V: (r, g, b)

3D Scenes and Shapes (as in NeRFs)

$r: (x, y, z, \theta, \phi)$, $V: (r, g, b, \sigma)$

Trajectories

$r: (q)_t^T$ generalized coordinates

V: utility function

✓ Continuous Representations

✓ "Infinite" Spatial Resolution

✓ Memory depends on signal complexity

✗ Not Analytically Tractable

Implicit Representation

Implicit Representations in Visual Computing

Shape reconstruction

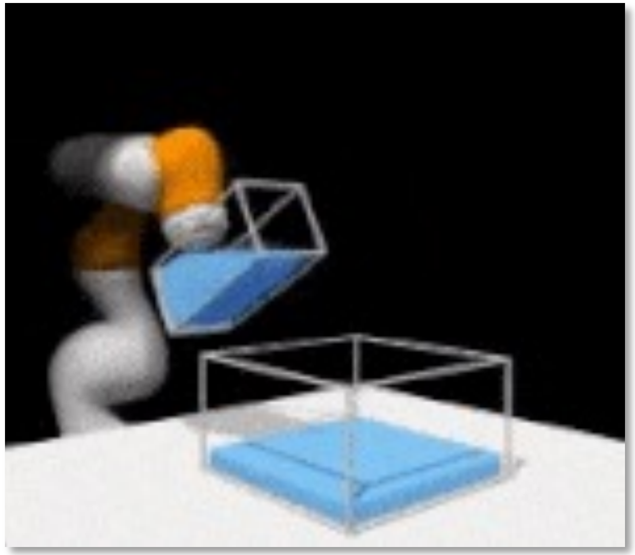
Rendering

Novel view synthesis

Occupancy Networks: Learning 3D Reconstruction in Function Space. In CVPR, 2019.
Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes. In CVPR, 2021.
NeRF: Representing Scenes as Neural Radance Fields for View Synthesis. In ECCV, 2020.

Implicit Neural Representations in Robotics

Grasp detection



Visuomotor control

Generalization in Manipulation

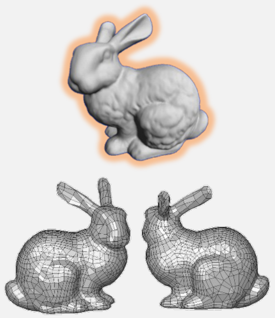
Synergies Between Affordance and Geometry: 6-DOF Grasp Detection via Implicit Representations. In RSS, 2021.
3D Neural Scene Representations for Visuomotor Control. In CoRL, 2021.
Neural Descriptor Fields: SE(3)-Equivariant Object Representations for Manipulation. In ICRA, 2022.

Robot Learning with Implicit Representations

Algorithmic Development (perception and control)
+ Improved Simulation for Contact-rich Manipulation

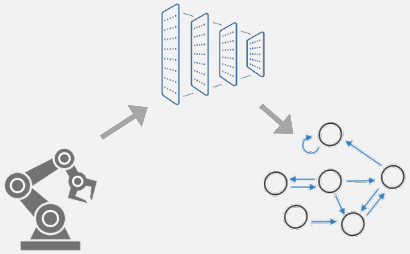
Perception

Objects & Poses



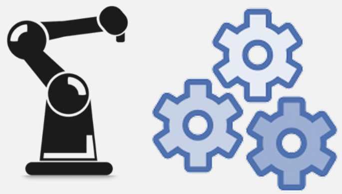
Action

Trajectories &
Value Functions



Simulation

Differentiable
contact sim



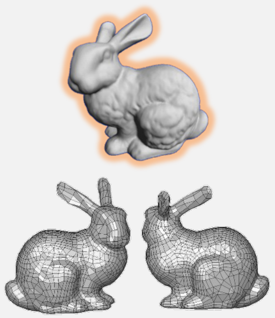
Work in progress

Robot Learning with Implicit Representations

Algorithmic Development (perception and control)
+ Improved Simulation for Contact-rich Manipulation

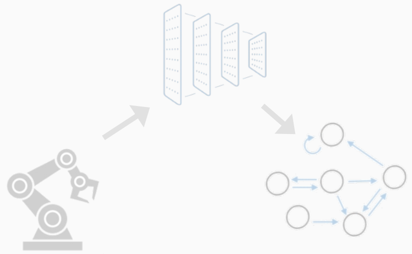
Perception

Objects & Poses



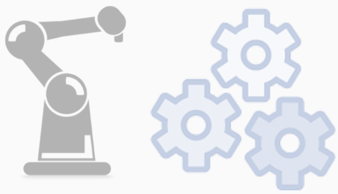
Action

Trajectories & Value Functions



Simulation

Differentiable contact sim

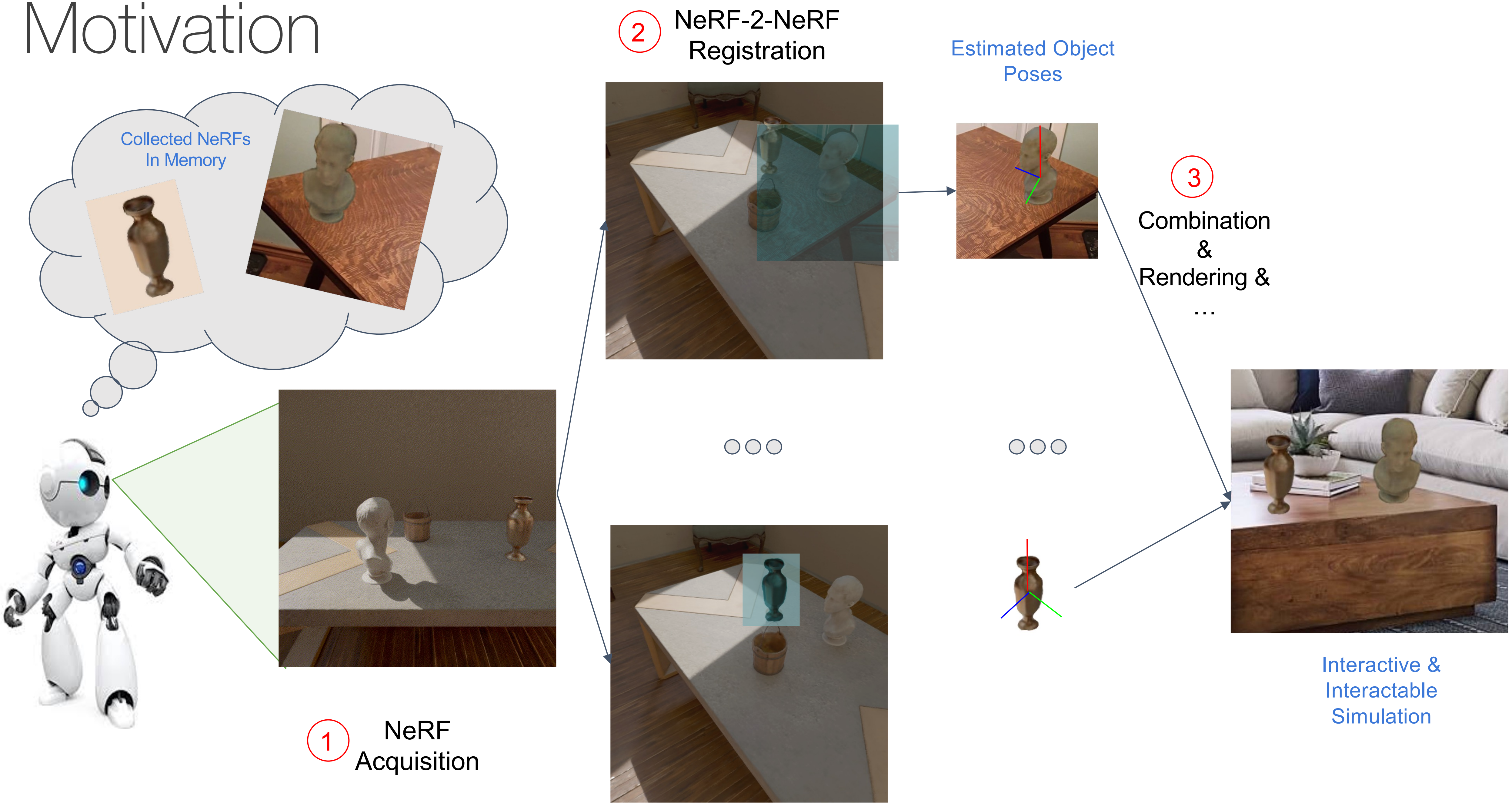


NERF 2 NERF

Registering Partially Overlapping NeRFs

Lily Goli, Daniel Rebain, Animesh Garg, Andrea Tagliasacchi

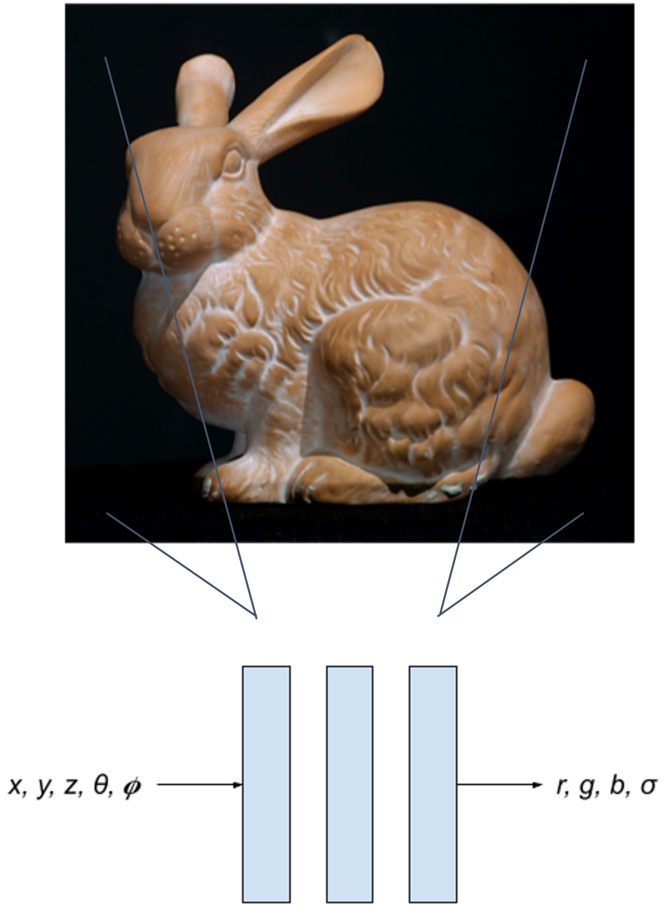
Motivation



Collected NeRFs
In Memory
2
NeRF-2-NeRF
Registration
Estimated Object
Poses
3
Combination
&
Rendering &
…
1
NeRF
Acquisition
Interactive &
Interactable
Simulation

What are Neural Radiance Fields (NeRFs)?

Training an MLP



Composition & Rendering

Rendering model for ray r(t) = o + td:

$$C \approx \sum_{i=1}^{N} T_i \alpha_i c_i$$

colors

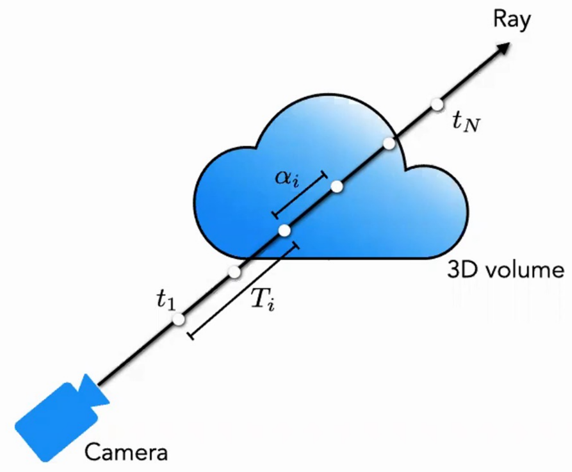
weights

How much light is blocked earlier along ray:

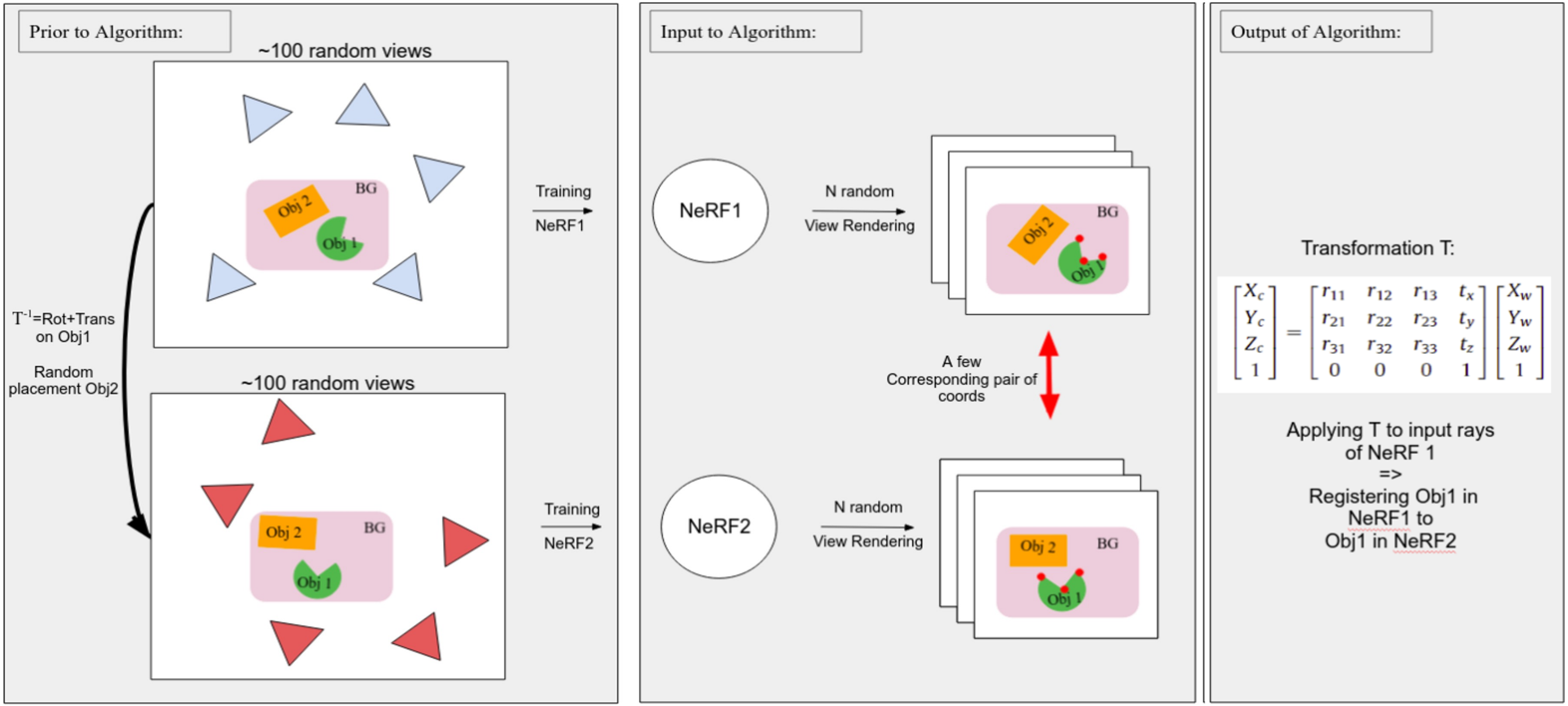
$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$

How much light is contributed by ray segment *i*:

$$\alpha_i = 1 - e^{-\sigma_i \delta t_i}$$



Registration Problem in NeRFs



Unsupervised Training to Find T - Objective Function?

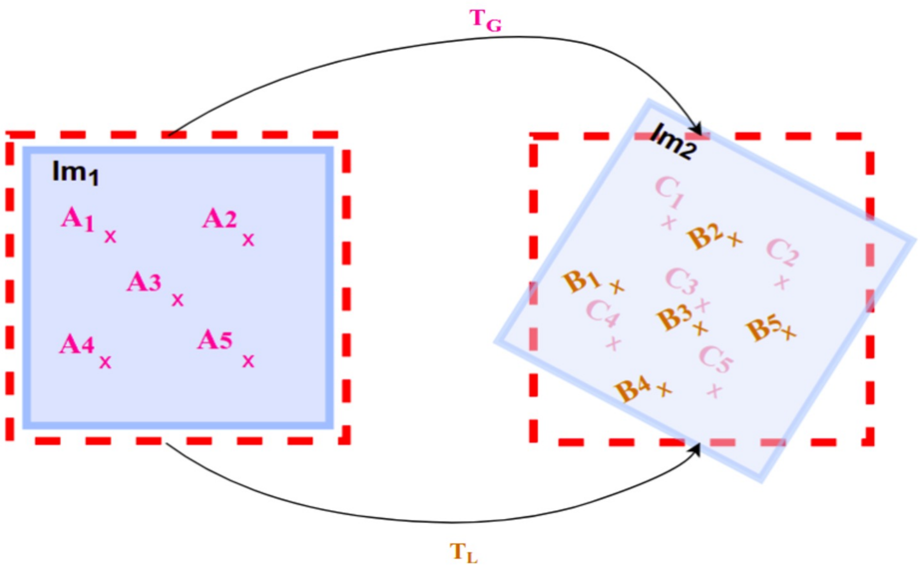
	View/RGB Difference		Correspondence Difference
Loss Function =	Error ($NeRF_1(T*R)$, $NeRF_2(R)$)	+	Distance between positions of corresponding point coordinates after applying T

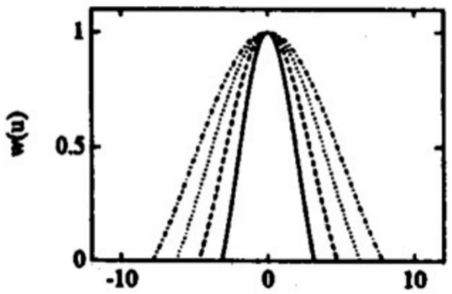
Challenges:

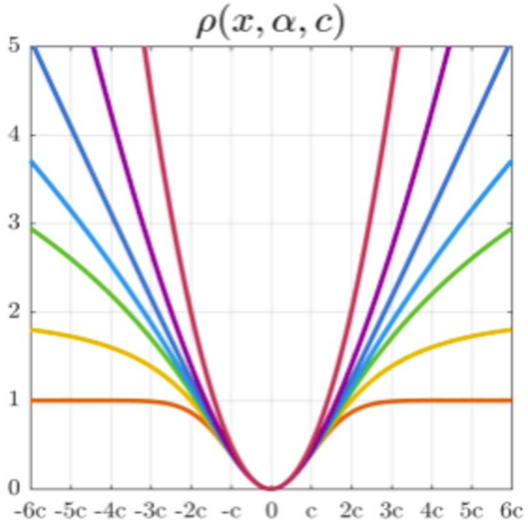
Even if learned T is optimal: Error between rendered images is NOT zero! The scenes are only *partially* overlapping.	=>	We need a robust function applied to MSE To make it more robust
Corresponding points lie in 2D space of rendered images Transformation T lies in 3D space	=>	We derive equivalent 3D Points using Triangulation

Focusing on First Loss Term (View Difference)

Robust Registration of 2D views. Modeling the problem in 2D setting:







- Delta will not be zero even if $T_L = T_G$, in some query points!

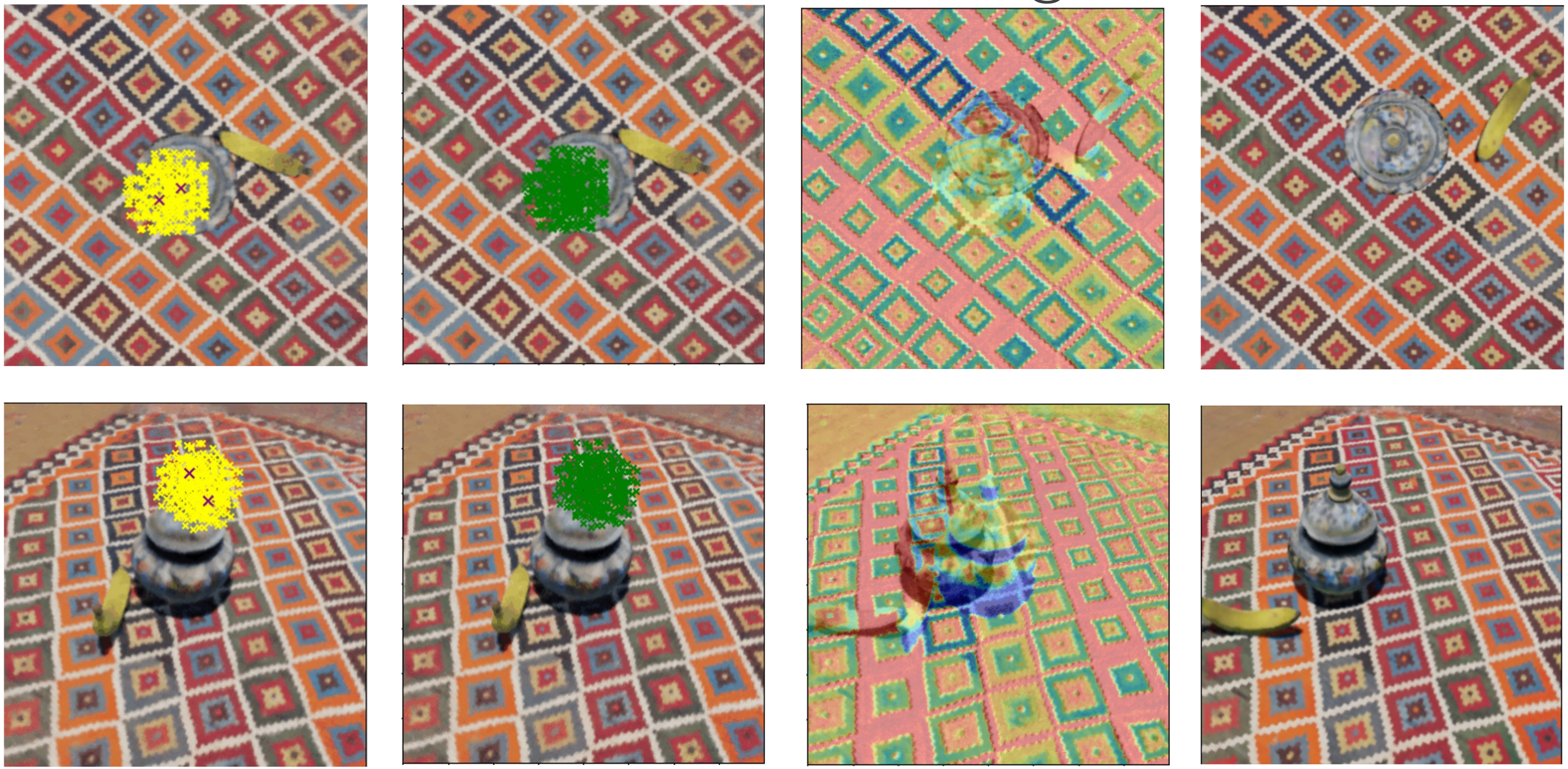
Just focus on object of interest -> many loss functions, mostly use manual thresholding

Barron CVPR 2019

Registration via Radiance Matching

Random view 1

Random view 2

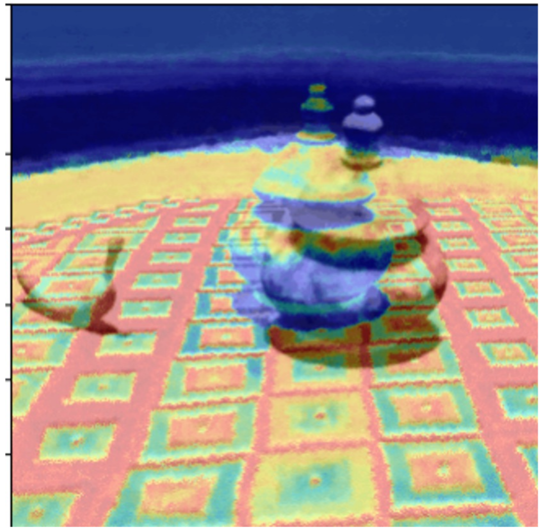


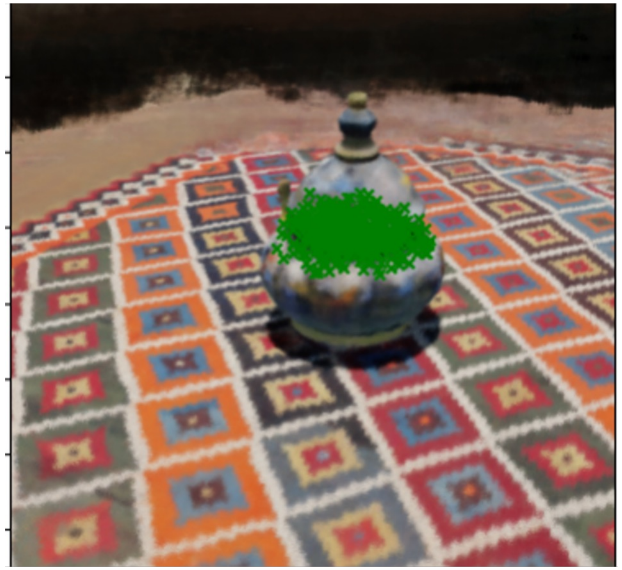
NeRF 1 without transform with sample points | NeRF 1 with transform with sample points | Overlap of fixed NeRF2 and moving NeRF1 | Fixed NeRF 2 (target)

Different Lightings (Failure Case)

If we use only radiance for registration,
then different lighting models on the object fail!

- Fix: Use Geometry features rather than radiance





Sampling in the moving NeRF

target view (uniformly lighter)

Geometry Network via Distillation

We train a 3 layer network supervised by:

How much light is blocked earlier along ray:

$$T_i = \prod_{j=1}^{i-1} (1 - \alpha_j)$$

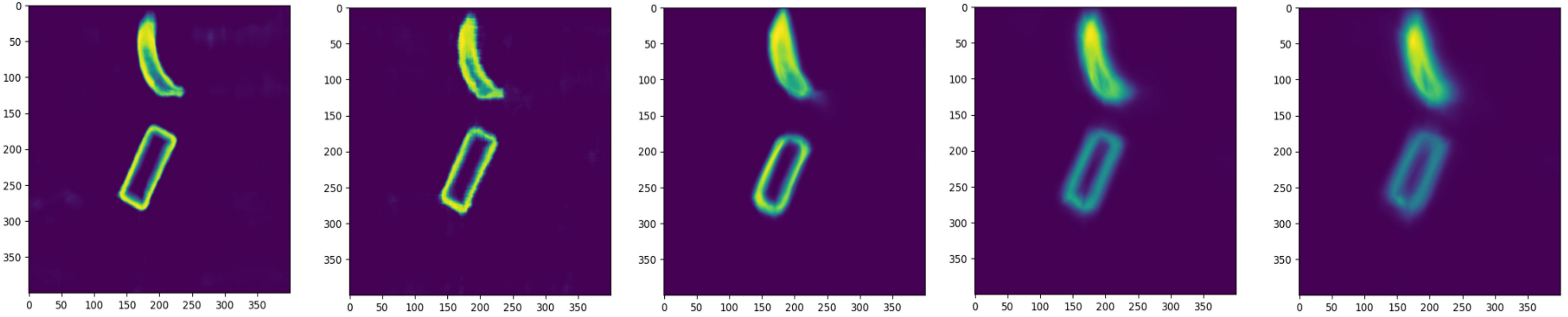
How much light is contributed by ray segment *i*:

$$\alpha_i = 1 - e^{-\sigma_i \delta t_i}$$

$$g(x) = max_{\delta}(\mathcal{F}(x, \delta))$$

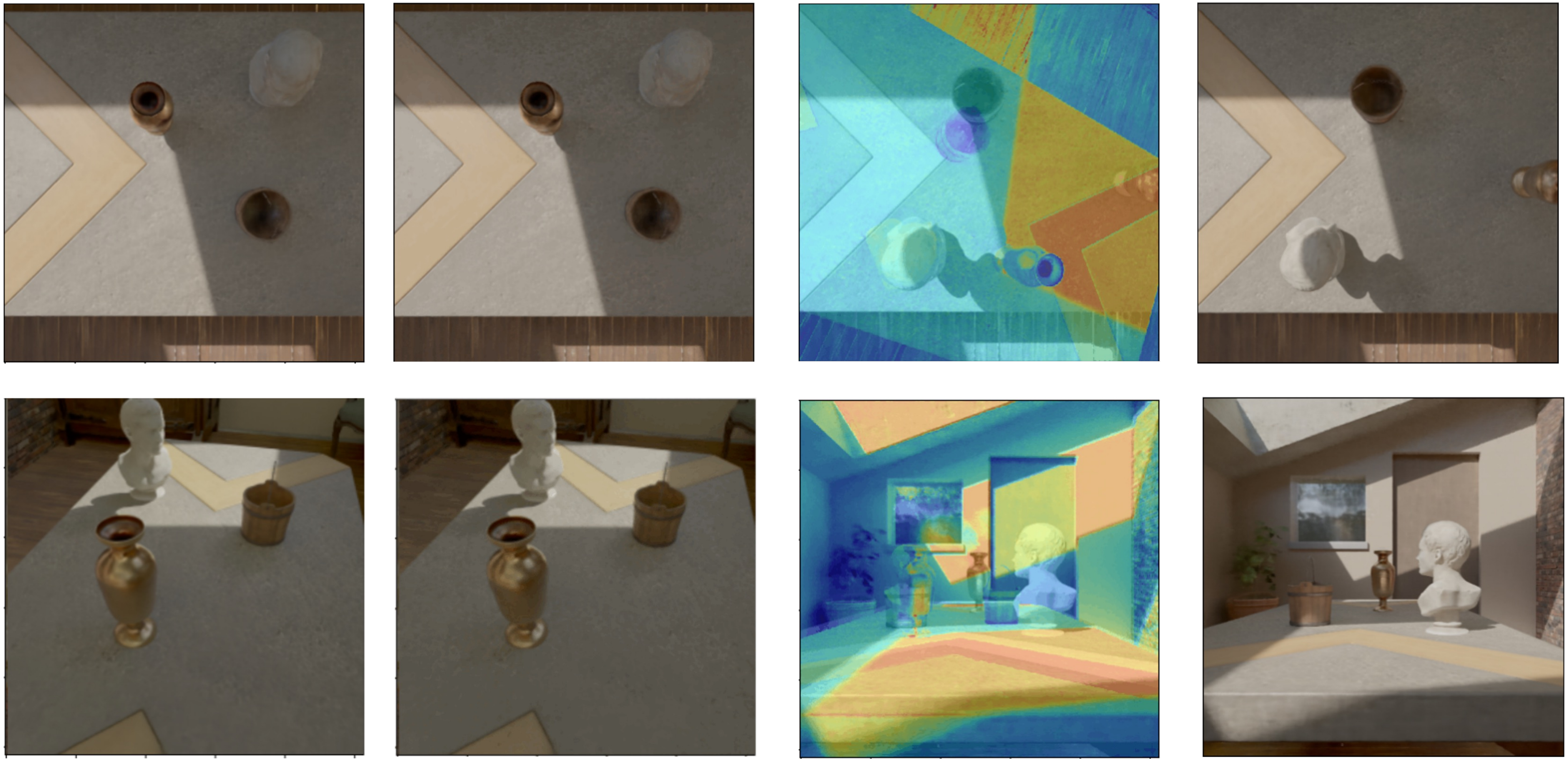
$$y \sim \mathcal{N}(x, \sigma)$$

$$f(y(x, \sigma)) = \frac{1}{n} \sum g(y)$$



Ground Truth | Sigma = 0 | Sigma = 0.005 | Sigma = 0.01 | Sigma = 0.02

Results



Random view 1

Random view 2

(moving) NeRF 1 - initial pose

NeRF 1 - registration iterations

Overlay of fixed NeRF 2 and moving NeRF 1

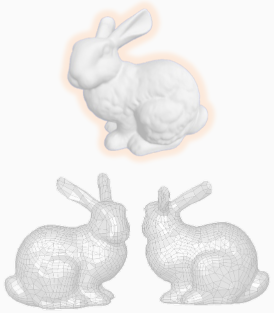
(fixed) NeRF 2 - target

Robot Learning with Implicit Representations

Algorithmic Development (perception and control)
+ Improved Simulation for Contact-rich Manipulation

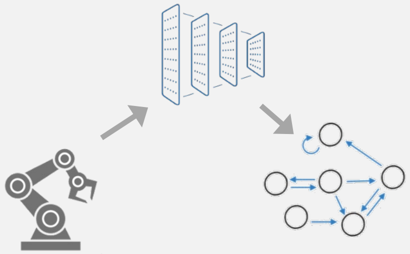
Perception

Objects & Poses



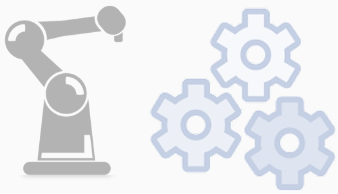
Action

Trajectories &
Value Functions



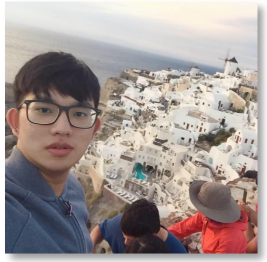
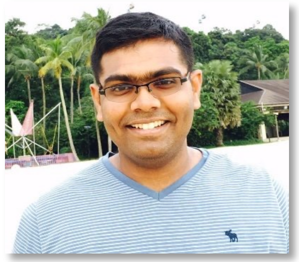
Simulation

Differentiable
contact sim



NEURAL MOTION FIELDS

Encoding Grasp Trajectories as Implicit Value Functions

Yun-Chun Chen, Adithya Murali, Bala Sundaralingam,
Wei Yang, Animesh Garg, Dieter Fox

Existing Grasping Methods

Grasp pose detection

6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In ICCV, 2019.

Existing Grasping Methods

Grasp pose detection

↓

Find inverse kinematic solutions

6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In ICCV, 2019.

Existing Grasping Methods

Grasp pose detection

↓

Find inverse kinematic solutions

↓

Plan a collision-free trajectory

6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In ICCV, 2019.

Existing Grasping Methods

Grasp pose detection

↓

Find inverse kinematic solutions

↓

Plan a collision-free trajectory

↓

Execute the open-loop trajectory

6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In ICCV, 2019.

Existing Grasping Methods

+ Table-top object grasping

+ Grasping in clutter

+ Bin-picking

Contact-GraspNet: Efficient 6-DOF Grasp Generation in Cluttered Scenes. In ICRA, 2021.

6-DOF Grasping for Target-driven Object Manipulation in Clutter. In ICRA, 2020.

6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In ICCV, 2019.

Existing Grasping Methods

+ Table-top object grasping

+ Grasping in clutter

+ Bin-picking

- Infer a finite discrete number of grasps

Contact-GraspNet: Efficient 6-DOF Grasp Generation in Cluttered Scenes. In ICRA, 2021.

6-DOF Grasping for Target-driven Object Manipulation in Clutter. In ICRA, 2020.

6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In ICCV, 2019.

Existing Grasping Methods

+ Table-top object grasping

+ Grasping in clutter

+ Bin-picking

- Infer a finite discrete number of grasps

Grasp affordances are a continuous manifold

Contact-GraspNet: Efficient 6-DOF Grasp Generation in Cluttered Scenes. In ICRA, 2021.

6-DOF Grasping for Target-driven Object Manipulation in Clutter. In ICRA, 2020.

6-DOF GraspNet: Variational Grasp Generation for Object Manipulation. In ICCV, 2019.

Neural Motion Fields

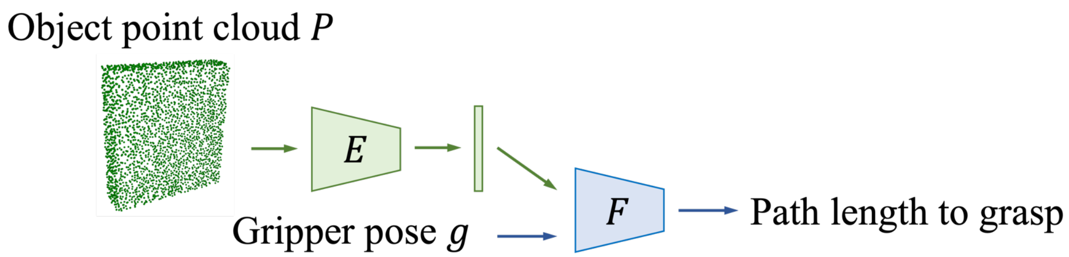
Goal:
Learn a value function that can be used to plan a trajectory for grasping

Neural Motion Fields

Goal:
Learn a value function that can be used to plan a trajectory for grasping

Value function:
Map a gripper pose to its path length to a grasp

$$\mathcal{L}_{\text{path-length}} = \|V_{\text{pred}}(g, P) - V_{\text{gt}}(g, P)\|_1$$

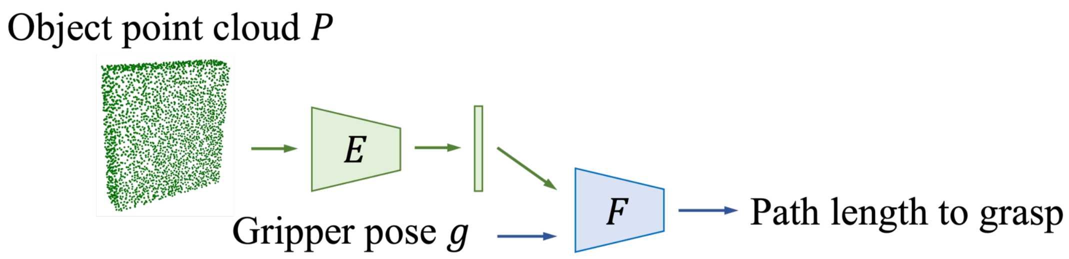


Neural Motion Fields

Goal:
Learn a value function that can be used to plan a trajectory for grasping

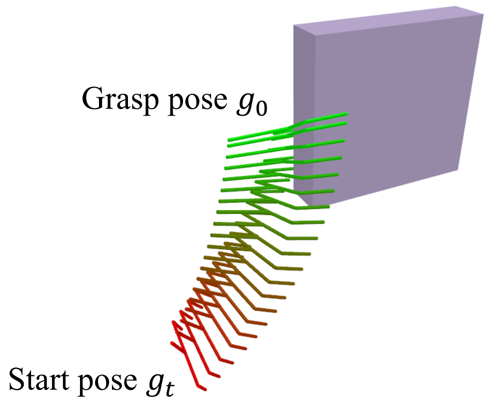
Value function:
Map a gripper pose to its path length to a grasp

$$\mathcal{L}_{\text{path-length}} = \|V_{\text{pred}}(g, P) - V_{\text{gt}}(g, P)\|_1$$



Gripper pose path length:

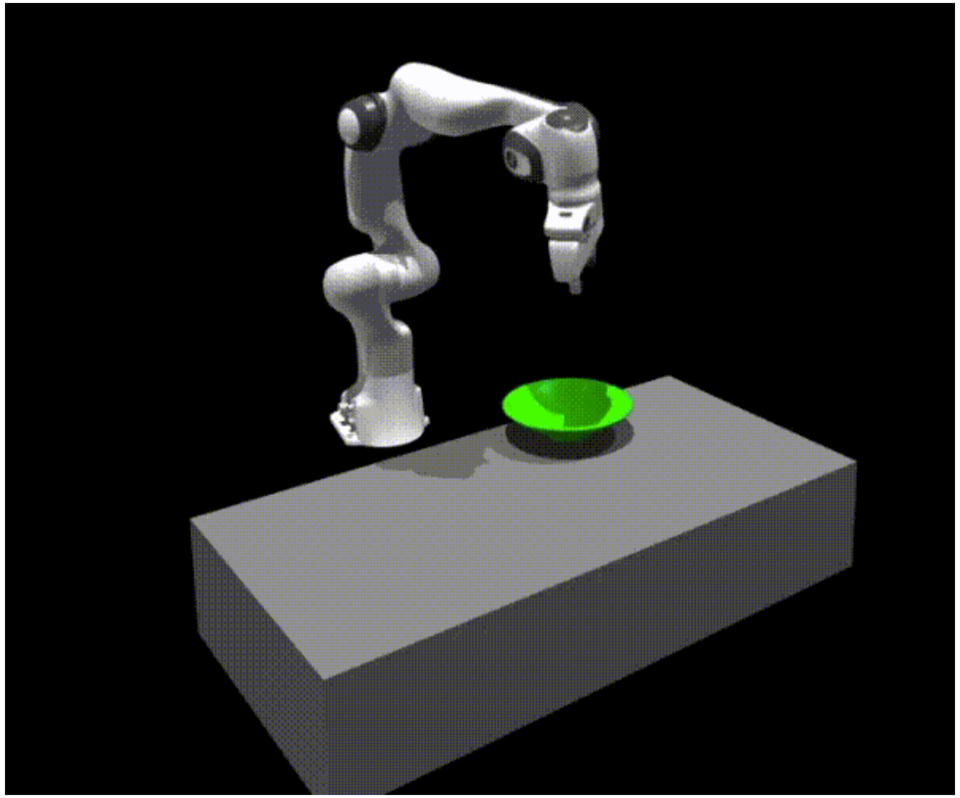
$$V(g_t) = \sum_{i=0}^{t-1} \frac{1}{m} \sum_{x \in M} \|(R_i x + T_i) - (R_{i+1} x + T_{i+1})\|$$



Grasp Motion Generation

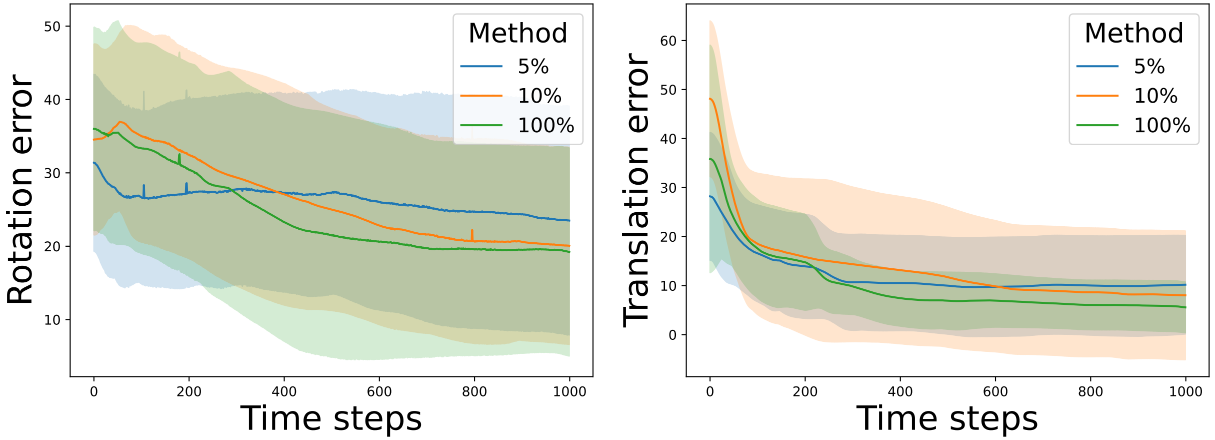
Query gripper poses and optimize the value function using a sampling-based MPC framework (MPPI)

$$\min_{\ddot{x}_{t\in[0,H]}} \quad \mathcal{C}_{\text{storm}}(q) + \mathcal{C}_{\text{grasp}}$$

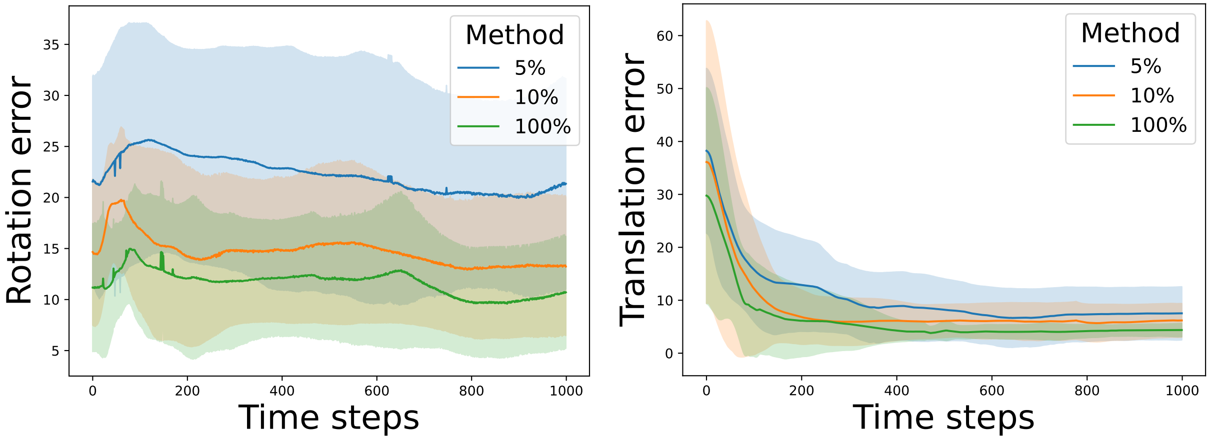


STORM: An Integrated Framework for Fast Joint-Space Model-Predictive Control for Reactive Manipulation. In CoRL, 2022.

Ablation Study on Number of Trajectories



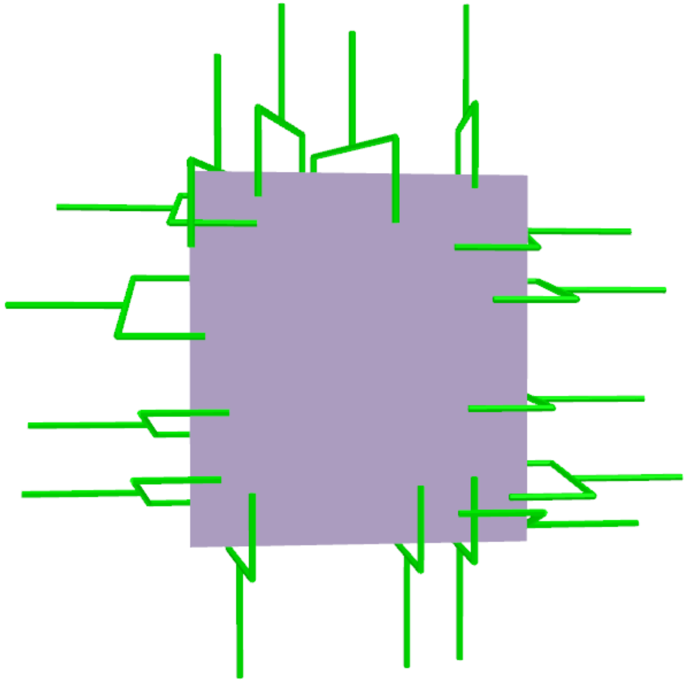
Static object poses



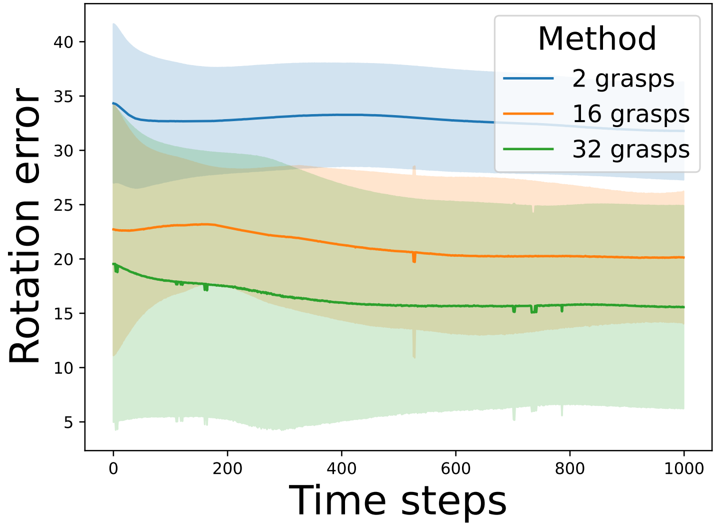
Dynamic object poses

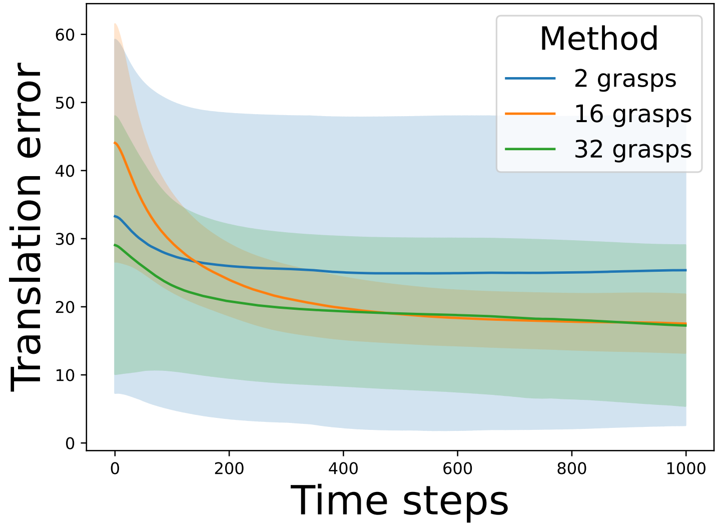
More data helps with fine-grained rotation error with non-stationary objects

Ablation Study on Number of Anchor Grasps



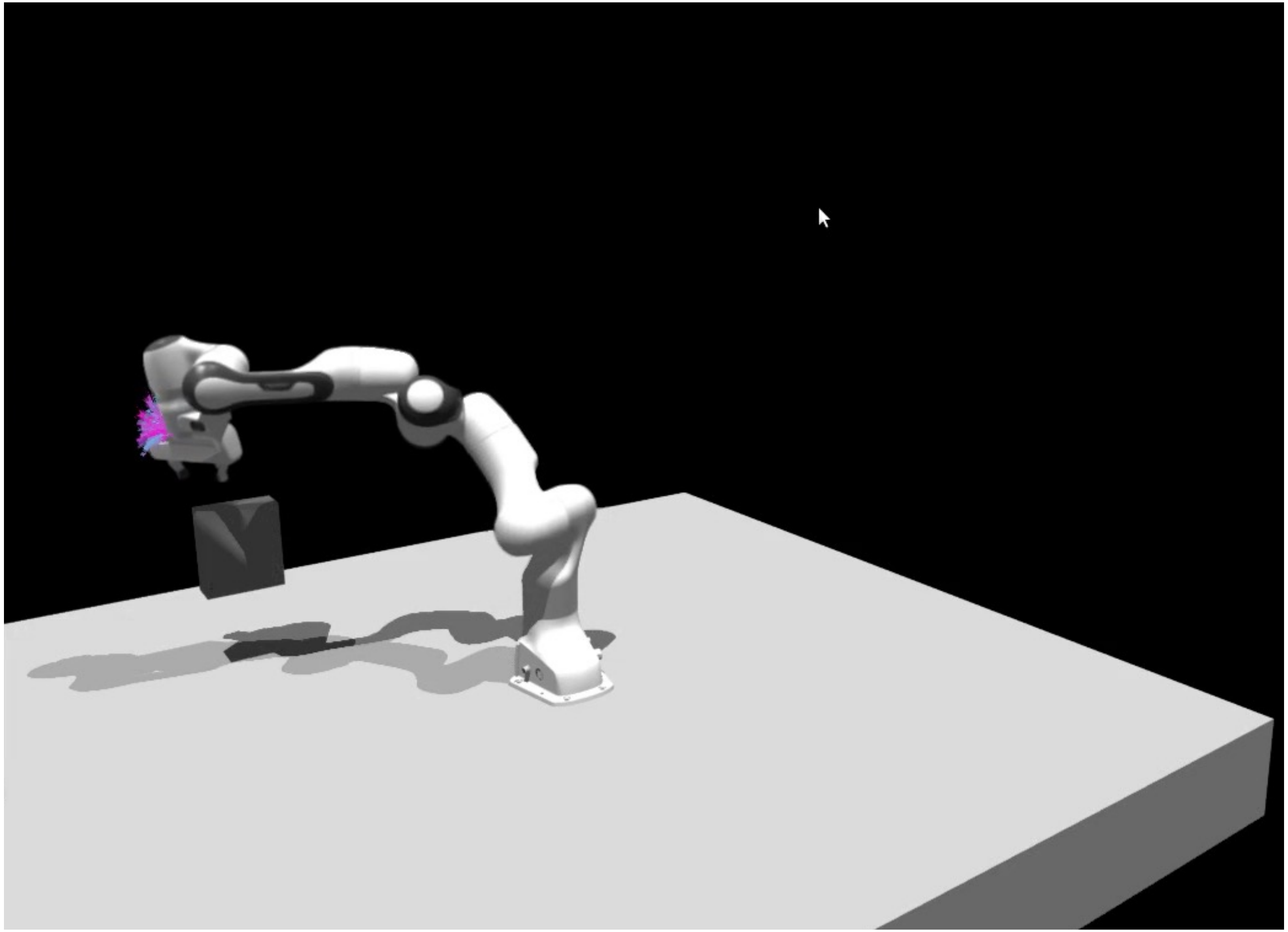
Anchor grasps





More data helps with snapping to multi-modal grasp prediction

Floating Object Demo

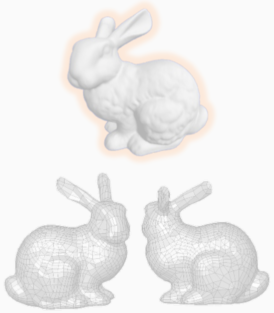


Robot Learning with Implicit Representations

Algorithmic Development (perception and control)
+ Improved Simulation for Contact-rich Manipulation

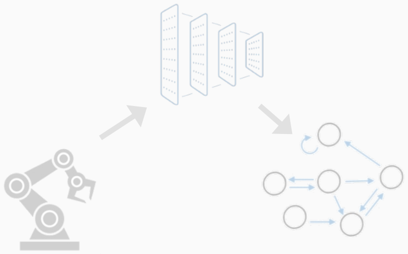
Perception

Objects & Poses



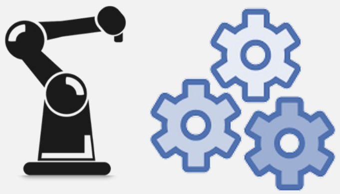
Action

Trajectories &
Value Functions



Simulation

Differentiable
contact sim



GRASP'D

Differentiable Contact-Rich Grasp Synthesis

Dylan Turpin, Liquan Wang, Eric Heiden, Yun-Chun Chen,
Miles Macklin, Stavros Tsogkas, Sven Dickinson, Animesh Garg

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity

   Only a fraction of possible contacts are active (in collision) at a given time.
   Inactive contacts have no gradient.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity

    Only a fraction of possible contacts are active (in collision) at a given time.
    Inactive contacts have no gradient.
    Can't follow gradient to create new contacts.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity
   Can't follow gradient to create new contacts.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity
   Can't follow gradient to create new contacts.
2. Local flatness
   Often compute ground-truth SDF from mesh.
   If closest point is on triangle face, surface normal gradient is 0.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity

   Can't follow gradient to create new contacts.

2. Local flatness

   Often compute ground-truth SDF from mesh.
   If closest point is on triangle face, surface normal gradient is 0.
   Can't follow gradient to improve contact normals.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity

   Can't follow gradient to create new contacts.

2. Local flatness

   Can't follow gradient to improve contact normals.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity
   Can't follow gradient to create new contacts.
2. Local flatness
   Can't follow gradient to improve contact normals.
3. Non-smooth object geometry
   Surface normals are often discontinuous (e.g., moving from one face of cube to another).

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity

    Can't follow gradient to create new contacts.

2. Local flatness

    Can't follow gradient to improve contact normals.

3. Non-smooth object geometry

    Surface normals are often discontinuous (e.g., moving from one face of cube to another).
    Can't follow gradient across non-smooth geometry.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity
   Can't follow gradient to create new contacts.
2. Local flatness
   Can't follow gradient to improve contact normals.
3. Non-smooth object geometry
   Can't follow gradient across non-smooth geometry.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity

   Can't follow gradient to create new contacts.

2. Local flatness

   Can't follow gradient to improve contact normals.

3. Non-smooth object geometry

   Can't follow gradient across non-smooth geometry.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges

1. Contact sparsity
   Can't follow gradient to create new contacts.
2. Local flatness
   Can't follow gradient to improve contact normals.
3. Non-smooth object geometry
   Can't follow gradient across non-smooth geometry.

So how can gradient-based optimization be possible?

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges & Proposed Solutions

1. Contact sparsity

   Can't follow gradient to create new contacts.

2. Local flatness

   Can't follow gradient to improve contact normals.

3. Non-smooth object geometry

   Can't follow gradient across non-smooth geometry.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges & Proposed Solutions

1. Contact sparsity → Leaky gradient

   Can't follow gradient to create new contacts, so allow gradient to *leak* through inactive contacts.

2. Local flatness

   Can't follow gradient to improve contact normals.

3. Non-smooth object geometry

   Can't follow gradient across non-smooth geometry.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges & Proposed Solutions

1. Contact sparsity → Leaky gradient
   Can't follow gradient to create new contacts, so allow gradient to *leak* through inactive contacts.
2. Local flatness → Phong SDF
   Can't follow gradient to improve contact normals, so borrow graphics techniques for smoothing.
3. Non-smooth object geometry
   Can't follow gradient across non-smooth geometry.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges & Proposed Solutions

1. Contact sparsity → Leaky gradient
   Can't follow gradient to create new contacts, so allow gradient to *leak* through inactive contacts.
2. Local flatness → Phong SDF
   Can't follow gradient to improve contact normals, so borrow graphics techniques for smoothing.
3. Non-smooth object geometry → SDF Dilation
   Can't follow gradient across non-smooth geometry, so consider the (smoothed, padded) radius r level-set.

Motivation

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

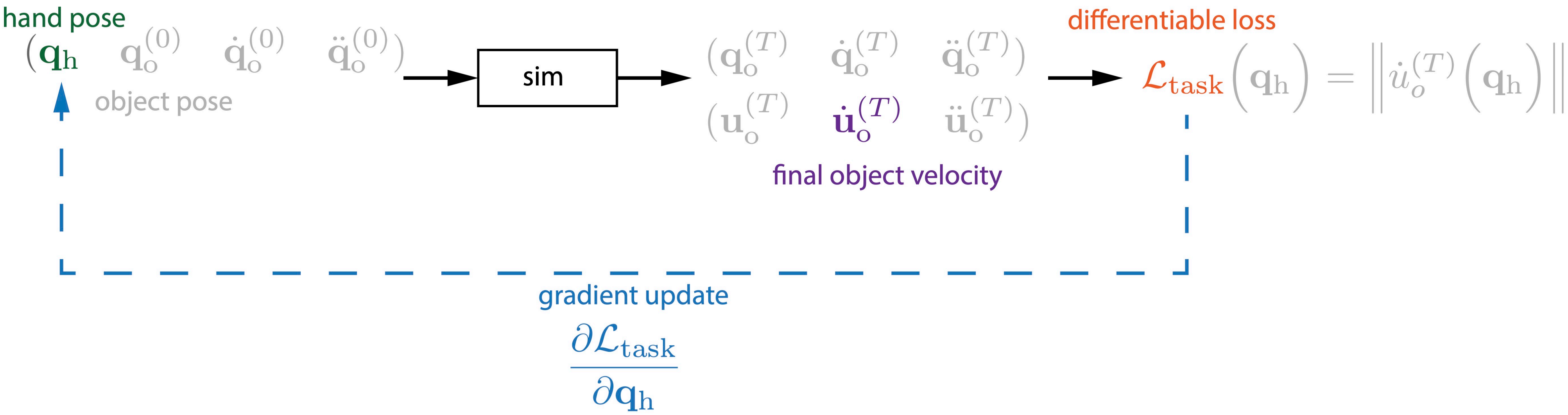
Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges & Proposed Solutions

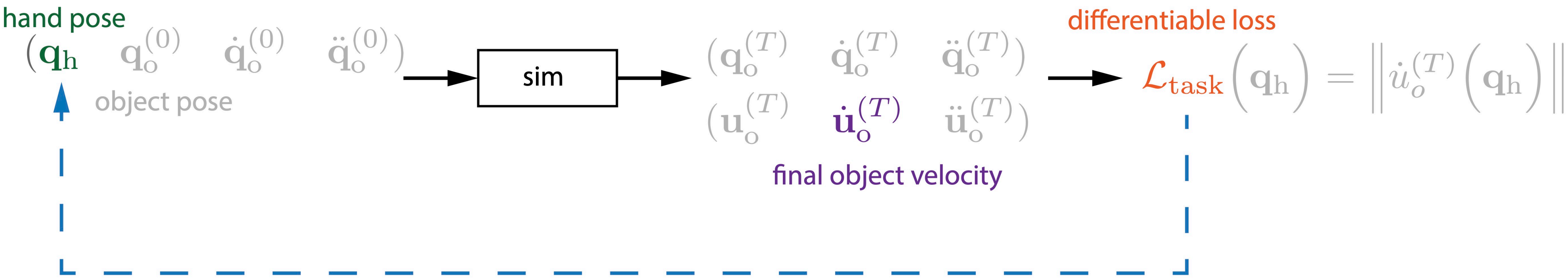
An example application: Generating *contact-rich* grasps for high-DOF human and robotic hands.

Like this… But how?

Challenge #1: Contact sparsity



Challenge #1: Contact sparsity



$$\boxed{\frac{\partial \mathcal{L}_{\mathrm{task}}}{\partial \mathbf{q}_{\mathrm{h}}} = \mathbf{0}}$$

Challenge #1: Contact sparsity

Proper gradient

$$\frac{\partial \left\| \mathbf{f}_n \right\|}{\partial \mathbf{q}} = \begin{cases} \boxed{k_n \frac{\partial \phi}{\partial \mathbf{q}} \quad \text{if } \phi(\mathbf{x}) < 0} \\ 0 \qquad\qquad \text{otherwise} \end{cases}$$

Non-zero if object SDF at contact
location is less than 0
(i.e., in collision)
and zero otherwise.

Challenge #1: Contact sparsity

Proper gradient

$$\frac{\partial \left\| \mathbf{f}_n \right\|}{\partial \mathbf{q}} = \begin{cases} k_n \frac{\partial \phi}{\partial \mathbf{q}} & \text{if } \phi(\mathbf{x}) < 0 \\ 0 & \text{otherwise} \end{cases}$$

Non-zero if object SDF at contact location is less than 0
(i.e., in collision)
and zero otherwise.

Leaky **gradient**

$$\frac{\partial \left\| \mathbf{f}_n \right\|}{\partial \mathbf{q}} := \begin{cases} k_n \frac{\partial \phi}{\partial \mathbf{q}} & \text{if } \phi(\mathbf{x}) < 0 \\ \boxed{\alpha k_n \frac{\partial \phi}{\partial \mathbf{q}} \quad \text{otherwise}} \end{cases}$$

Gradient when not in collision is just scaled down by alpha.

Challenge #2: Local flatness

SDF ground truth is often computed from a mesh.

But surface normal is constant on faces,
so contact normal (computed as positional derivative of SDF) has 0 gradient.

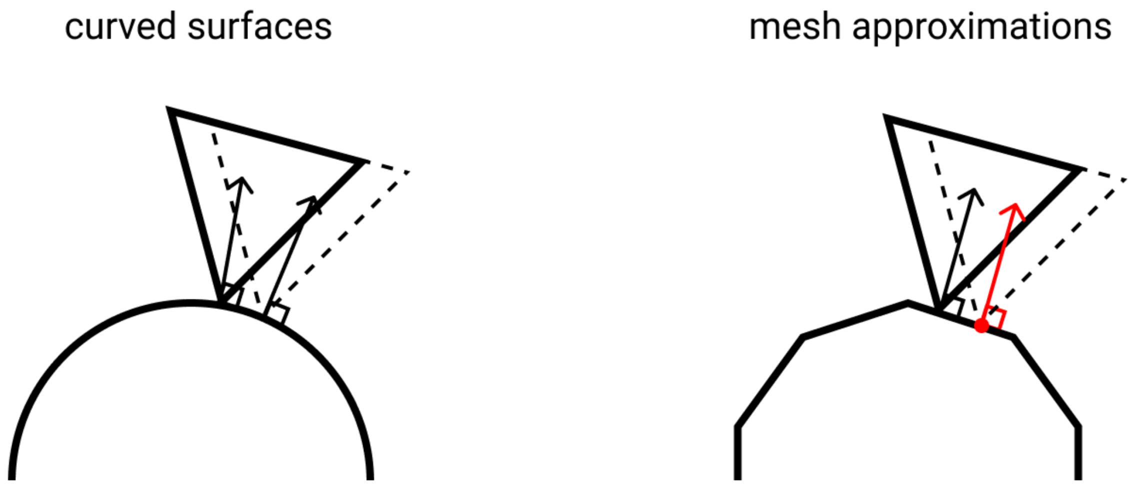


Figure from Werling, K., Omens, D., Lee, J., Exarchos, I., & Liu, C. K. Fast and Feature-Complete Differentiable Physics for Articulated Rigid Bodies with Contact.

Challenge #2: Local flatness

SDF ground truth is often computed from a mesh.

But surface normal is constant on faces,
so contact normal (computed as positional derivative of SDF) has 0 gradient.

Many possible solutions!

We use one simple trick by analogy to ray-tracing: Phong tessellation.

Challenge #2: Local flatness

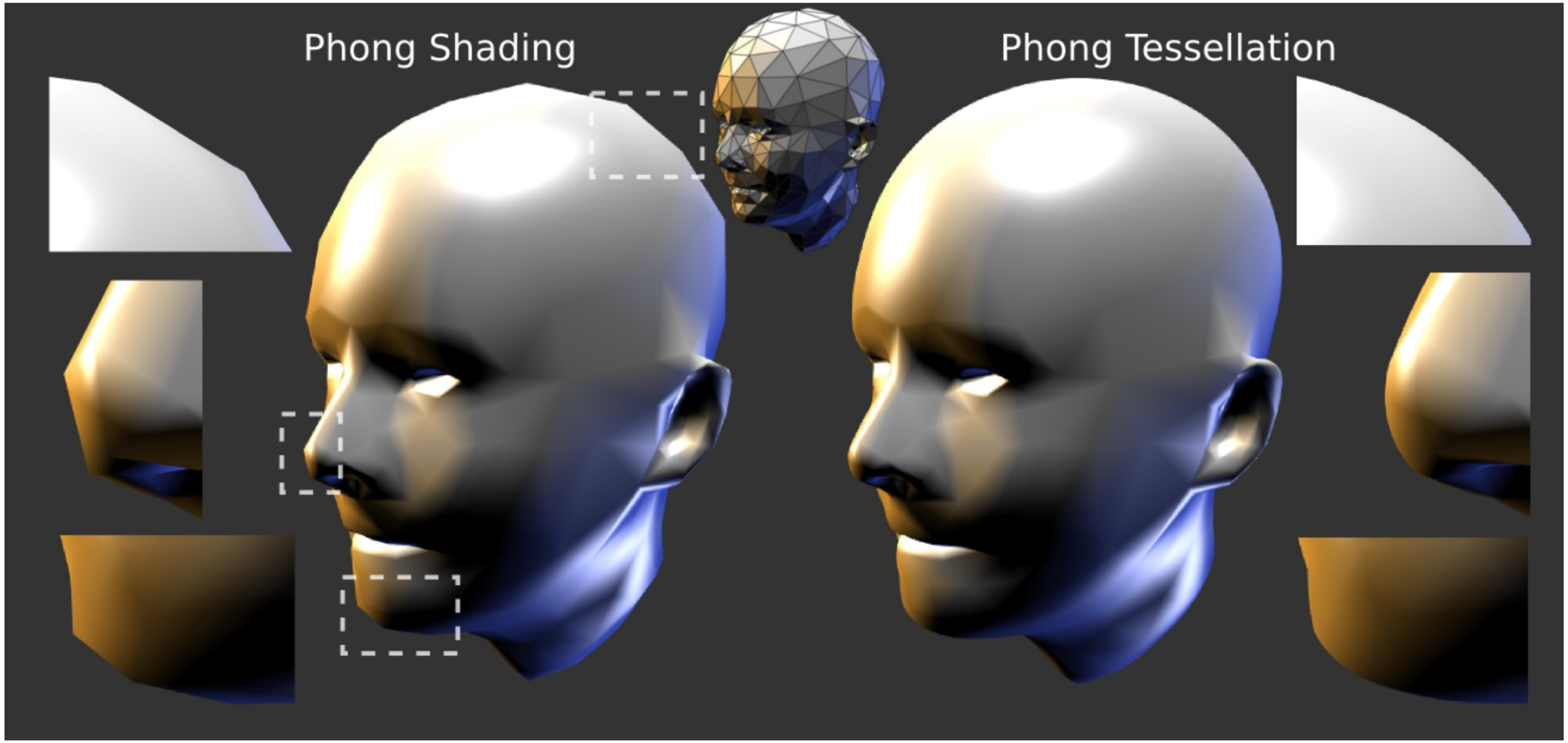


Figure from Phong Tessellation T Boubekeur, M Alexa ACM Transactions on Graphics 27 (5)

Challenge #3: Non-smooth geometry

Easy to optimize over surface of a spherical cow (⚽),
but most aren’t so smooth (🐄).

Challenge #3: Non-smooth geometry

Easy to optimize over surface of a spherical cow (⚽️),
but most aren't so smooth (🐄).

Discontinuities in surface normals
↓
discontinuities in contact normals
↓
discontinuities in their gradients with respect to contact positions.

Challenge #3: Non-smooth geometry

Luckily for us… SDFs are easy to smooth.

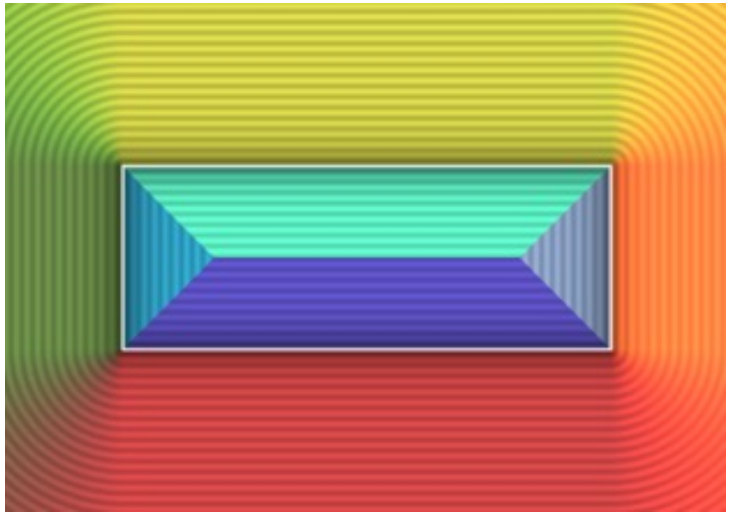


Figure from inigo quilez interior SDFs (2020)

Challenge #3: Non-smooth geometry

Luckily for us… SDFs are easy to smooth.

Instead of the sdf=0 level set,
consider the sdf=*r* for some *r*>0.

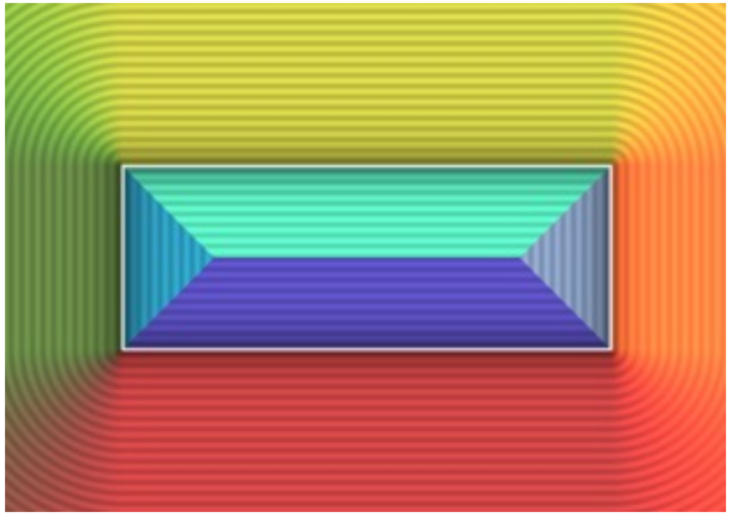


Figure from inigo quilez interior SDFs (2020)

Challenge #3: Non-smooth geometry

Luckily for us… SDFs are easy to smooth.

Instead of the sdf=0 level set,
consider the sdf=*r* for some *r*>0.

Adjust towards true surface (*r*=0)
as optimization progresses.

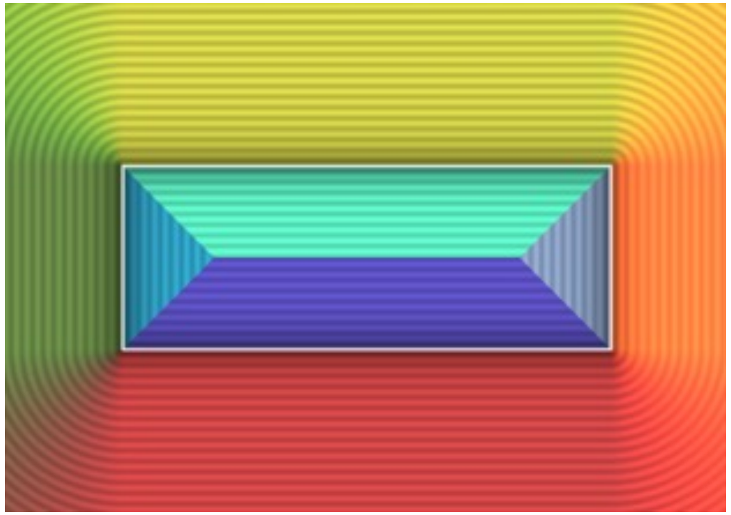


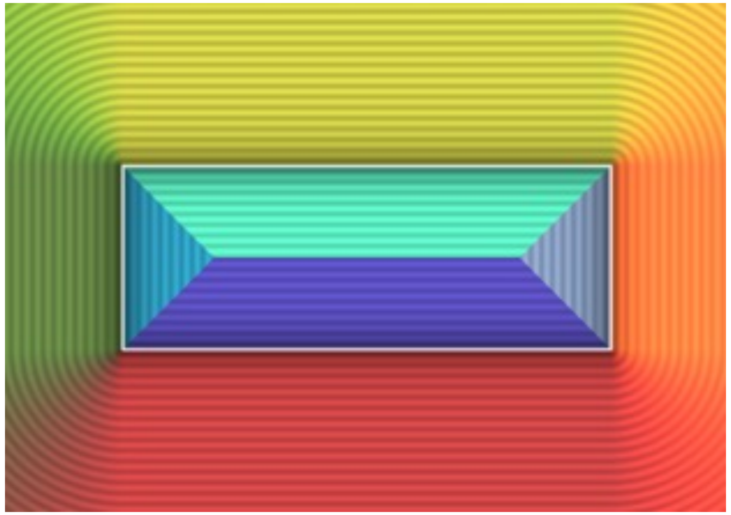
Figure from [inigo quilez](https://iquilezles.org) interior SDFs (2020)

Challenge #3: Non-smooth geometry

Luckily for us… SDFs are easy to smooth.

Instead of the sdf=0 level set,
consider the sdf=r for some r>0.

Adjust towards true surface (r=0)
as optimization progresses.



For robotic grasping:
Hand pre-shapes as if grasping larger version of same object.

Figure from inigo quilez interior SDFs (2020)

Challenge #3: Non-smooth geometry

Does not help concave corners.

Future work: Is there a better transform?

Figure from [inigo quilez](#) interior SDFs (2020)

Grasps from the ObMan dataset [*]

- Simplifying assumptions

[*] Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., & Schmid, C. (2019). Learning joint reconstruction of hands and manipulated objects. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11807-11816).

Grasps from the ObMan dataset [*]

- Simplifying assumptions → Bias towards fingertip only grasps

[*] Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., & Schmid, C. (2019). Learning joint reconstruction of hands and manipulated objects. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11807-11816).

Grasps from the ObMan dataset [*]

- Simplifying assumptions → Bias towards fingertip only grasps

↓

less contact

[*] Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., & Schmid, C. (2019). Learning joint reconstruction of hands and manipulated objects. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11807-11816).

Grasps from the ObMan dataset [*]

- Simplifying assumptions → Bias towards fingertip only grasps

less contact

less stable

Less contact = less friction.

[*] Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., & Schmid, C. (2019). Learning joint reconstruction of hands and manipulated objects. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11807-11816).

Grasps from the ObMan dataset [*]

- Simplifying assumptions → Bias towards fingertip only grasps

less contact

less stable — Less contact = less friction.

less plausible — Human grasping is contact-rich.

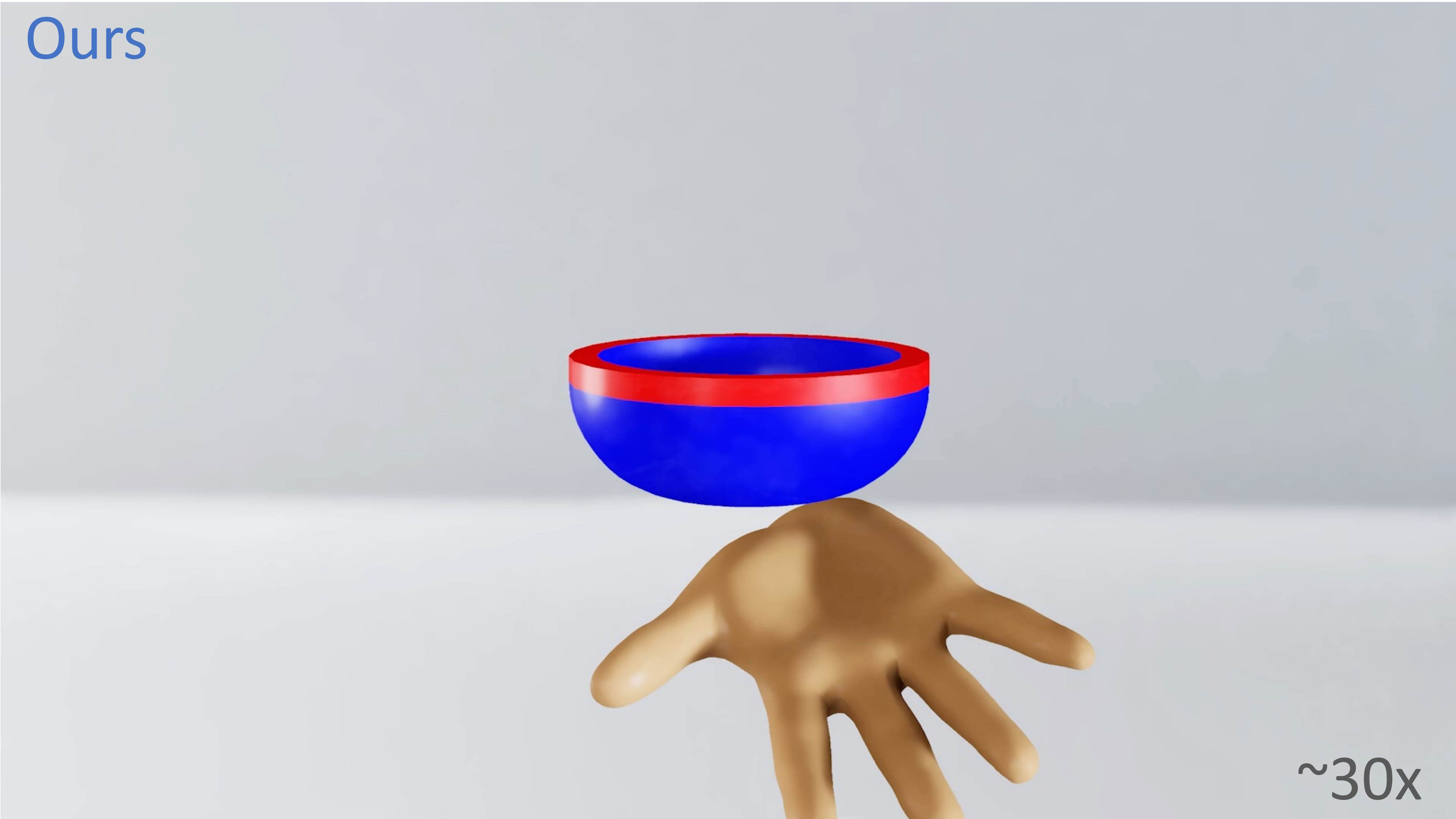
[*] Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M. J., Laptev, I., & Schmid, C. (2019). Learning joint reconstruction of hands and manipulated objects. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11807-11816).

ObMan

Ours

~30x

ObMan


Ours
~30x


ObMan
Ours

Ours

~30x

Grasp'D: Take away

Goal: Make *SDF-based* contact forces friendly to *gradient-based* optimization.

Why? Planning in high-dimensional contact-rich scenarios,
e.g., robotic grasping and manipulation with multi-finger hands.

Challenges & Proposed Solutions

1. Contact sparsity → Leaky gradient
2. Local flatness → Phong SDF
3. Non-smooth object geometry → SDF Dilation

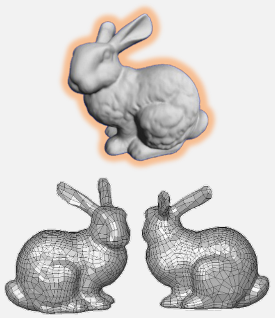
An example application: Generating contact-rich human & robotic grasps.

Robot Learning with Implicit Representations

Algorithmic Development (perception and control)
+ Improved Simulation for Contact-rich Manipulation

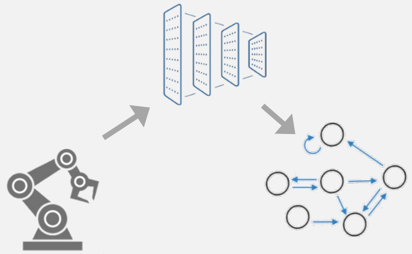
Perception

Objects & Poses



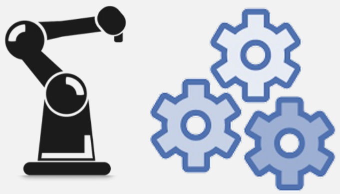
Action

Trajectories &
Value Functions



Simulation

Differentiable
contact sim

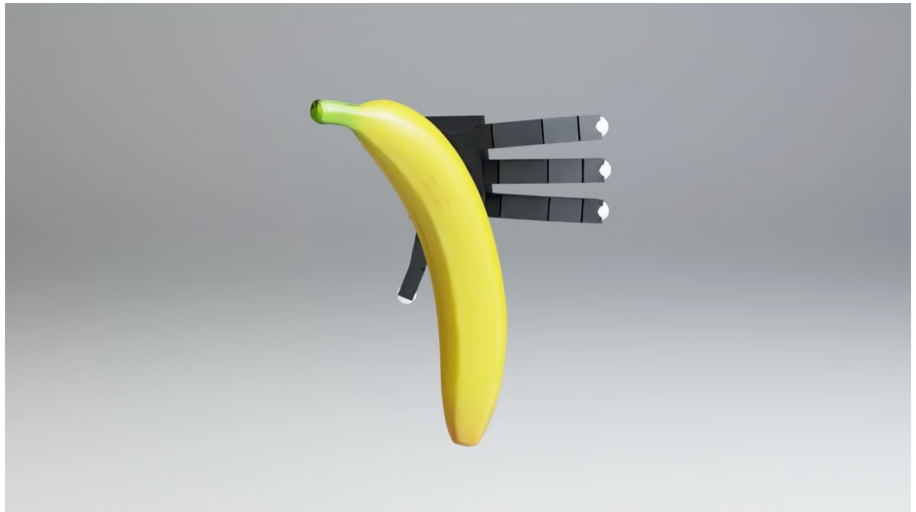


INR: object oriented state representations+ planning and control
Key challenge: pre-training and generalization

Work in progress

Robot Learning with Implicit Representations

Perception, Action, and Simulation



Animesh Garg

garg@cs.toronto.edu | @animesh_garg

People AI & Robots pair

UNIVERSITY OF TORONTO

VECTOR INSTITUTE

NVIDIA